Processor-in-the-Loop Simulation and Multivariable Control System Design for Pressurizer System in Nuclear Power Plants

Amal A. Sheta [1], Elsayed H. Ali [1], Refaat M. Fikry [1], Tarek A. Mahmoud [2], Sayed M. El-Araby [1], Mohammed I. Mahmoud [2]

[1] Engineering Department, Nuclear Research Centre, Egyptian Atomic Energy Authority, Egypt.
[2] Automatic Control and Industrial Electronics Department, Faculty of Electronic Engineering, Menoufya University.

E-mails: med_rob_5a2010@yahoo.com, sayedmahdy@yahoo.com , eng_refaat@yahoo.com, Tarek.Mohamed@el-eng.menofia.edu.eg, selaraby@netscape.net, m_i_mahmoud@yahoo.com.

Abstract

One of the famous and energy-efficient types of nuclear power plants is the pressurized water reactor (PWR). The pressuriser (PZR) system is essential for the safe functioning of a PWR as it regulates primary coolant pressure variations during normal operations and maintains pressure within designated boundaries. The PZR unit in the PWR is a Multi-Input Multi-Output (MIMO) and nonlinear dynamic system. Accordingly, the control strategies of the PZR system are too complex. The pressure and water level of PZR directly affect the performance operation of the PWR load power. Therefore, an adaptive PID-based fuzzy logic controller is proposed in this paper to control both the PZR pressure and water level with the derivation of their control signals. The PZR model with the proposed controllers is implemented using the MATLAB/Simulink environment. Also, this research aims to drop the proposed control systems in practice firmware and ensure low cost and less memory consumption of the processor. Therefore, the proposed controllers are validated by the processor in the loop (PIL) test using the STM32F407 discovery kit. The results of different PZR operation scenarios show the improvement of PZR water level performance and pressure using the proposed adaptive PID controllers, with a remarkable coincidence between their simulation and PIL implementation response.

Keywords: Nuclear Power plant, PZR system, Adaptive PID Control, Fuzzy Control, PIL.

I. INTRODUCTION

In a Pressurized Water Reactor (PWR), the pressurizer (PZR) is an essential part of the primary circuit (PC). It is used during the steady-state operation to maintain the reactor control system (RCS) pressure at the predetermined setpoint value. In contrast, during the transient operations, it controls the resultant variation of the pressure within allowed tolerances [1, 2]. PZR compensates for the positive and negative fluctuation due to the transient change of load power. The PZR maintains water and steam saturation within the balance to preserve the PC pressure. During this regular quick operation, the reactor does not stop working. The PZR, through the different reactor accidents, prevents the primary loop system from overpressure with keeping its integrity. The quick changes in the primary loop system water volume are absorbed by the PZR, which works as a buffer container of the primary loop [3]. According to these reasons, the pressure and water level of the PZR control system are significant for the safe operation of PWRs. The PWRs pressure control is realized by controlling electrical heaters and spray valves. Therefore, the PZR comprises a spray system, a unit for an electrical heater, a group of safety valves, and measuring devices. Accordingly, the PC pressure is decreased by spraying the steam area with a low-temperature coolant that flows through the spray system, composed primarily of spray valves. The electrical heater unit is also used to increase the PC pressure by heating the coolant of the PZR.

It is important to model the PZR's thermal-hydraulic activity during transients to achieve control objectives [4-6]. PZR behaviour is exceedingly complex since it is nonlinear, has significant inertia, is time-variant, and has unpredictable open-loop dynamic properties [7]. Previously, alternative mathematical models with simpler assumptions were utilised for thermal-hydraulic simulation of the PZR in earlier investigations. For example [8, 9] developed a non-equilibrium three region model for the PZR system on the mass and energy conservation laws. In [10] indicate that PC variations influence the two-region thermal-hydraulic model. In [11] presented the four-region model with fewer assumptions related to the other models. A non-equilibrium control-oriented model for PZR dynamic behaviour modelling in PWRs has been developed in [12]. In addition, the multi-region model is made up of three layers, each of which is divided into multiple control volumes (regions) [13].

Several forms of works use specialized controllers for the PZR system modelling to enhance the efficiency of pressure control during the secure operation of the PWR. Some researchers have used different controllers to the PZR, either linear, artificial intelligent, or nonlinear. The fuzzy controller was developed by [14] for pressure control of a PZR model based on artificial neural networks (ANNs). In [9] suggested a fuzzy-logic-based approach for evaluating the constant pressure setpoint in the PZR pressure control scheme. The PZR pressure control method is presented by [15], based on an adaptive prediction algorithm. The Particle Swarm Optimization (PSO) algorithm optimized PID controller gains for power-level control of a standard PWR in 2018 by [4]. The water level of the linearized PZR model was controlled by a fuzzy PID controller in 2011 [6]. In 2013, a single neural PID controller was also proposed to control the PZR pressure [6]. The Fuzzy-PID controller was first used in the nuclear reactor power control system in 2013, and it was tuned using the Genetic Algorithm (GA) [16]. The evolutionary algorithm was used to investigate multi-objective optimization of PID controller parameters for the PZR pressure and level control systems [17]. In 2019, a linearized non-equilibrium three-region PZR model was developed to control the PZR pressure and water level using PID controllers [8]**.**

The nonlinearity and complexity of the PZR mathematical models steer all researchers to concentrate their efforts towards linearized models or models based on artificial intelligence (AI) techniques for designing pressure and level control systems. However, these controllers do not reflect the reality and effectiveness evaluation for the performance operation of PZR. Moreover, it should be underlined that most of the advanced and intelligent controllers provided also reflect the reality since the pressure and level control mechanisms of the PZR are worked independently (SISO system). Also, these two systems depend on each other during shrink and swell phenomena in the PZR operation.

There are different control algorithms to the PZR system. It is not possible to apply one of these algorithms on PZR pratically directilly, Building prototypes to test and evaluate the performance of a system could thus be really essential. But this method entails long-term process design with increasing expenses and occasionally damaging dangers for researchers. Platforms for offline simulation are widely used as validation and verification tools to depict systems in order to investigate their behavior and related phenomena.

Between simulation and hardware implementation, there are several phases. For example, embedded software often follows the V-cycle development approach [18, 19]; X-In-The-Loop is a model-driven technique mostly used for testing. The four tiers of testing configurations offered in these tests are MIL (Model-In-The-Loop), SIL (Software-In-The-Loop), PIL (Processor-In-The-Loop), and HIL (Hardware-In-The-Loop)[20]. Each configuration level contributes to the development process by closing the gap between the mathematical model and the firmware executing on a standalone microprocessor platform.

The MIL configuration is the starting point for a model-driven system approach, which includes basic simulation to examine the controller model as well as the simulated plant model. The main idea behind MIL is to create and validate some test cases for your model on a high-precision float-pointing arithmetic platform, allowing you to examine how your model acts and how good it is on that platform. The idea of this stage is to produce the reference output test results for your controller for the next ones. Any mathematical controller model's defects may be rapidly found and corrected, hence the MIL phase comes early in the course of development.

Executable code running in fixed-point arithmetic on the same computer platform replaces the model used in the MIL test in Software-In--Loop (SIL). This stage helps the system developer spot any bad choices about memory size very fast. Usually, these two operations are carried out on the one integrated platform of a PC. Normally, these two procedures are completed on a PC's single integrated platform. If your controller system will run on hardware with a floating point dedicated unit in its CPU data-path, the SIL step can be bypassed.

The next phase is the PIL (Processor-In-The-Loop) test, which goes beyond the PC platform[21, 22]. This step incorporates some hardware capabilities that enable the control algorithm to run in more realistic settings. In PIL, the target processor is a non-real-time environment, and communication with external processors is accomplished by utilising certain capabilities installed on the host PC in a simulated integrated environment. PIL requires drivers to link the computer platform to pertinent hardware. After that, the created object code is downloaded most usually to an off-the-shelf evaluation board equipped with the target processor, where it is integrated with other test-management tools. The simulation tool on the PC then uses a serial communication channel to talk to the downloaded programme. The PIL simulation sends test values over a serial link to the firmware installed on the evaluation board's CPU, and then waits for a response from the processor using the same or another communication channel.

The HIL test is used to test the software real-time functioning, which cannot be done in the PIL. Although this may appear to be a limitation at first glance, it really allows us to split the simulation problem into two sections, each of which may be checked before we know for sure that the controller firmware will operate correctly on the independent processor platform. PIL facilitates the evaluation of compiler optimisations on a non-real-time execution platform, utilising an off-the-shelf processor architecture for firmware execution. HIL is characterised as the ultimate phase in the development of an embedded controller system. Hardware-in-the-loop (HIL) simulation must incorporate electrical emulation of sensors and actuators on a real-time target platform prior to validating the controller with actual plant sensors and actuators. These electrical emulations serve as a connection between the plant simulation and the embedded system under evaluation on the same platform. The plant simulation modifies the values of each electrically reproduced sensor, which is interpreted by the embedded system under evaluation, providing real-time feedback analogous to the conditions the controller will encounter prior to its installation for managing a real plant.

The main goal of this work is to employ an effective model of the PZR system combining various characteristics of PZR dynamics and the parameters influenced by the PC with fit for control applications. Thus, concurrently created and implemented for the nonlinear two-region PZR model is the derivation of multiple-input and multiple-output (MIMO) pressure and level control systems utilising standard PID controller.

However, the constant gains of the PID controller are not suitable for the case of set point variation. Therefore, the adaptive PID (APID) based fuzzy controller is applied for pressure and level control system for the nonlinear PZR model. Furthermore, the selection of an APID algorithm is based on its tracking speed, steady-state performance, and ability to be implemented on an integrated board, ensuring

great robustness. Also, each controller PIL for the PZR unit is implemented and evaluated for different PZR operation scenarios. Finally, the PZR is implemented using the MATLAB/Simulink environment, and setpoint changes are carried out to study the robustness of the PZR model with the proposed controllers. Simulation results indicate that the proposed APID controller performs better than the conventional one. Finally, STM32F407 discovery kit Platform is used to execute the PIL simulation to implement the proposed controllers connected with the PZR model. Some drivers must communicate the hardware with the computer platform (PIL). The generated object code is linked with test-management functionality and then downloaded. A serial link is then used to communicate the downloaded software with the simulation tool that runs on the laptop machine. Finally, a serial communication protocol is used to connect the simulation model of the PZR with the system of the STM32F407 discovery kit. Furthermore, the tools used to build and validate the software must meet the DO-178C standard. As a result, Matlab/Simulink is employed as a qualified tool [23, 24].

The rest of this paper is organized as follows. Section 2 represents the description of the pressurizer model, while Section 3 deduces the pressure and water level of PZR control strategies. Section 4 studies the adaptive PID-based fuzzy logic control of pressure and water level of the PZR system. The proposed adaptive PID controller is applied for different MIMO pressurizer system operation scenarios illustrated in section 5. Finally, section 6 shows the PIL implementation of the adaptive PID controller of pressure and water level of the PZR system for different scenarios and compared with simulation results.

II. PRESSURIZER MODEL DESCRIPTION

Although many mathematical models have described the pressurizer operation, including three, four, and multi-region models [2, 25, 26], the mathematical models did not deduce the relation for describing the surge line flow rate of the PZR with the primary circuit. Therefore, here a two-region model was used through which we were able to prove a mathematical relationship for describing the surge line equation. Furthermore, based on the nonlinear two-region model equations of a PZR by [10], the model has been demonstrated to have good performance in terms of predicting dynamic behaviours of PWR pressurizers during a wide range of operational transients, which can be expressed by a set of ordinary differential equations in (1). The PZR in the mathematical modelling consists of two regions, the steam, and water region, connected to the PC heat pipe section through the surge line (hot leg). Fig (1) illustrates the different PZR areas and divisions, illustrating the unit's other thermodynamic processes.

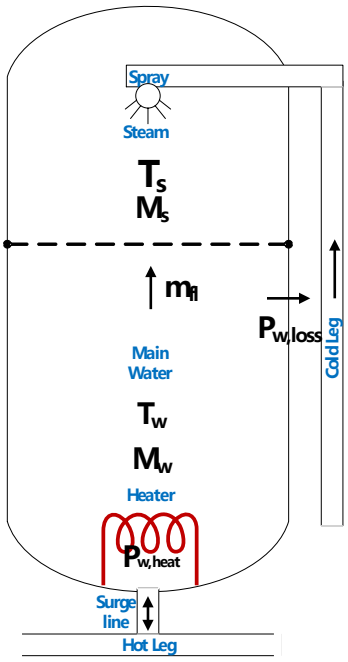


Fig. 1. Structure of the two region PZR System in PWRs.

$$
\begin{cases}
\dfrac{dM_w}{dt} = m_{su} - m_s \\
\dfrac{dM_s}{dt} = m_{s,PRZ} \\
\dfrac{dM_{pc}}{dt} = m_{in} - m_{out} - m_{su} \\
\dfrac{d(h_s M_s)}{dt} = M_s\dfrac{d(h_s)}{dt} + h_s\dfrac{dM_s}{dt} = h_s m_s + V_s\dfrac{dP}{dt} + K_{PZR}(T_w - T_s) - P_{s,\,loss} \\
\dfrac{d(h_w M_w)}{dt} = M_w\dfrac{d(h_w)}{dt} + h_w\dfrac{dM_w}{dt} = h_{w,\,H.L.}m_{su} - h_s m_s + V_w\dfrac{dP}{dt} + K_{PZR}(T_w - T_s) + P_{w,\,heat} \\
\dfrac{dT_s}{dt} = \dfrac{1}{c_{p,s}M_s}\left[-m_s h_s + V_s\dfrac{dP}{dt} + K_{PZR}(T_w - T_s) - P_{s,\,loss}\right] \\
\dfrac{dT_w}{dt} = \dfrac{1}{c_{p,w}M_w}[(h_{w,\,H.L.} - h_w)m_{su} - (h_{s*} - h_w)m_s + V_w\dfrac{dP}{dt} + K_{PZR}(T_w - T_s) + P_{w,\,heat} - P \\
p_{PZR} = p^*(T_{w,\,PZR}) = a_0 - a_1 T_{w,\,PZR} + a_2 T^2_{w,PZR} \\
L_{PZR}(M_{pc}, T_{pc}) = \dfrac{1}{A_{PZR}}\left(\dfrac{M_{pc}}{\rho(T_{pc})} - V_{pc}\right)
\end{cases}
\quad (1)
$$

Table (1): ***Parameters and Symbols [10].***

Parameter	***Symbol***	***Parameter***	***Symbol***
Power of reactor	Wr	The water level of PZR	l_{PZR}
PC overall mass	M_{pc}	Pressure of PZR	p_{PZR}
PC Average temperature	T_{pc}	Specific heat of water at 325°	$c_{p,pc}$
PC inlet mass flow rate	m_{in}	Heat loss	W_{loss}
PC outlet mass flow rate	m_{out}	Water mass	M_w
The water temperature of the hot leg	$T_{pc,hl}$	Steam mass	M_s
The water temperature of the cold leg	$T_{pc,cl}$	Cross-section of a vessel of PZR	A_{PZR}
The inlet temperature of the PC	$T_{pc,i}$	Vessel volume of PZR	$V_{PZR,vessel}$
Specific heat at 282°C	$c_{p,pc}$	The pressure of the saturated vapour	$p_{*,T}(T)$
Heat transfer Coefficient	$k_{t,sg}$	Coefficients for quadratic approximation	a_0 a_1 a_2
Steam temperature	T_s		
The water temperature of PZR	T_w		
Heat loss of PC	$W_{loss,pc}$	Water density	$\mathcal{P}$ (T)
PC nominal volume of water	$V_{pc,0}$	Coefficients for quadratic approximation	c_0 c_1 c_2
Power of heaters	P_{heat}		
PC nominal mass of water	$M_{pc,0}$		
Maximum spray flowrate	W_{spray}^{max}	Power of variable heater	q_{h}^{max}
Temperature difference	Δ		

The change of load in PWRs requires a corresponding change in the pressure, either increasing or decreasing. Also, the heater and spray operation scenarios are related to pressure, and accordingly, the pressure is associated with the surge line of water in/out of the PZR. The surge flow rate m_{su} is derived by equation (2). The estimated formula of surge flow rate depends on the average temperature of PC coolant, temperature of the hot leg, and inlet/outlet mass flow rates of the coolant in/out PC. Also, it depends on the reactor power, steam generator temperature, the inlet temperature of the PC, and the PC's power loss. These parameters affect the PZR operation as they increase or decrease the PZR pressure of a PWR, reflected in the change of load power.

$$
\begin{aligned}
m_{su} &= \frac{(m_{in} - 31m_{out})M_{pc} - V^{\circ}_{pc} c_1 (m_{in} T_{pc},}{M_{pc}} \\
&+ \frac{V^{\circ}_{pc} c_1}{c_{p,PC} M_{PC}} (Q_{loss,pc} - W_R + 4K_{T,SG} (T_{PC} \\
&- T_{SG})) \\
&+ \left[\frac{V^{\circ}_{pc} c_1 (m_{in} - m_{out}) - 2V^{\circ}_{pc} c_2 m_{in} T_{PC}}{M_{PC}} \right. \quad (2) \\
&+ \left. \frac{2V^{\circ}_{pc} c_2 (m_{in} - m_{out}) T_{PC}}{M_{PC}} \right. \\
&+ \frac{2V^{\circ}_{pc} c_2 (Q_{loss,pc} + 4K_{T,SG} (T_{PC} - T_{SG}) - W_R - 30m_{out}}{c_{p,PC} M_{PC}} \\
&* T_{PC} \\
&+ \left[\frac{2V^{\circ}_{pc} c_2 T_{PC} m_{out} + V^{\circ}_{pc} c_1 m_{out}}{M_{PC}} \right] \\
&* T_{PC,HL}
\end{aligned}
$$

Consequently, the PZR model is divided into three stages (inputs, states, and outputs). The first stage of the model is the input stage used to build all PZR inputs. The intermediate stage consists of five states, which are M_s that represents the mass of the steam inside the PZR, while the next state is M_w that means the water mass change inside the pressurizer. The water mass of the primary circuit M_{pc} is represented by the third subsystem. The fourth and the fifth subsystems are T_w and T_s that means the temperature of PZR water and temperature of its steam, respectively.

III. PRESSURIZER CONTROL SYSTEM

The flow rate of spray and the power of electrical heaters are the two most common techniques to manage the pressure of a PZR. The adjustment is carried out in response to a control signal based on an input pressure error signal. As a result, the controller is commonly referred to as the pressure error controller. Furthermore, to maintain the water level of PZR at its operating point, the inlet flow rate should be regulated. Fig (2) indicates the block diagram for controlling both the PZR pressure and level by keeping a constant output flow rate using a cascaded flow rate error controller system.

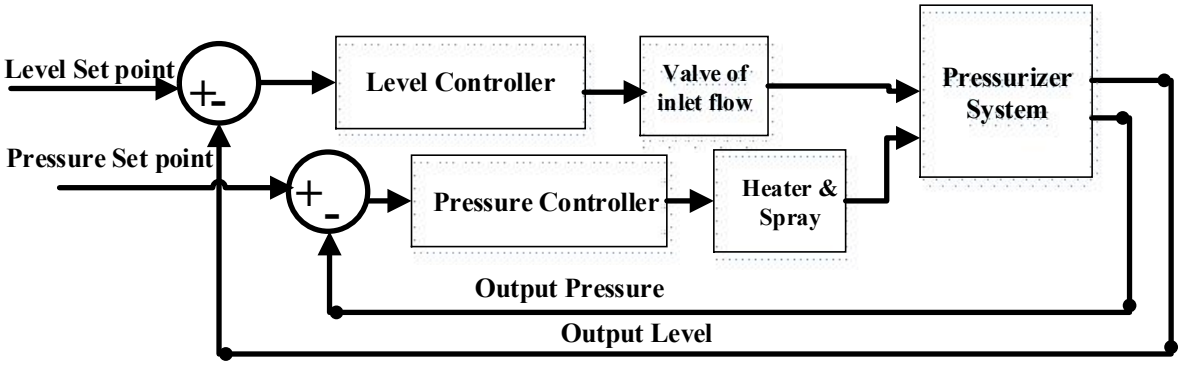


Fig. 2. General block diagram of MIMO pressurizer control system.

The electrical heaters of a PZR are classified into two groups. The first group consists of one bank of variable heaters, while the other group contains several other banks of backup on/off heaters. The applied voltage of variable heaters is made variable to vary the heaters' output heat over a set range of pressure. Throughout the steady-state conditions, the equilibrium heat balance within the PZR is maintained by variable heaters. They provide a maximum output of heat for the case in which the pressure has a considerable decrease from the determined set point. Also, in that case, the backup heaters are turned on. All heaters are turned off for the other case in which the PZR pressure is increased higher than this point. Also, the spray valves are opened over a fixed pressure range to allow the condensation of steam by cooler water, and hence the pressure returns to its average values.

For the case of a significant system response determined by the controller in a process control system, the controllers of pressure, level, and flow rate error should tune their parameters to get a satisfactory performance of the control of the PZR. Fig (3) illustrates a schematic diagram for PZR pressure control [8].

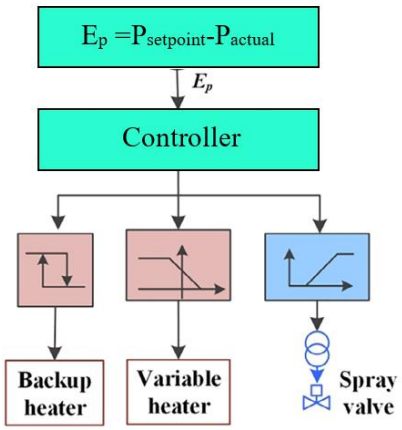


Fig. 3. Pressure control of the PZR.

For the other case of controlling the water level of PZR, it is noted that the level decreases when there is a leakage in RCS or a decrease in its average temperature. The decrease in water level is reflected in the increase of steam volume (out surge), the reduction of steam temperature, and the decrease in steam pressure and system pressure. On the other hand, the decreasing steam volume (in surge) is reflected in the increase of water level, the increasing steam temperature, and both steam and system pressure. The PZR level is adjusted by controlling the charging pumps flow rate, where the case of a decrease in the level is treated by increasing the inlet flow rate in such a way that exceeds the outlet

flow. The other point of increasing the level is treated by decreasing the inlet flow to be lower than the outlet one. The error signal of the water level is used to control the inlet flow rate of the PC. The control objective is to design dynamic controllers for the level and pressure of the PZR system.

The control signal (i.e., inlet mass flow rate to the PC) that is required to maintain the level of water inside the PZR at the desired reference value can be derived as in equation (3):

$$
\begin{aligned}
m_{in} &= \frac{1}{\left(\frac{1}{\mathcal{P}\left(T_{PC}\right)} - \frac{\left(T_{PC,I} - T_{PC}\right)\left(c_1 - 2c_2 T_{PC}\right)}{\left(\mathcal{P}\left(T_{PC}\right)\right)^2}\right)}\Big(A_{PZR}\Big[\frac{d}{} \\
&+ k_p\left(l_{PZR,ref} - l_{PZR}\right) \\
&+ k_i\int_0^t \left(l_{PZR,ref}(t) - l_{PZR}(t)\right) dt\Big] \\
&+ \frac{m_{su}}{\mathcal{P}\left(T_{PC}\right)} + \frac{m^{o}_{out}}{\mathcal{P}\left(T_{PC}\right)} \\
&+ \frac{\left(c_1 - 2c_2 T_{PC}\right)}{c_{p,PC}\left(\mathcal{P}\left(T_{PC}\right)\right)^2}\left(c_{p,pc} m^{o}_{out}\Delta^{o} + W_R - 4W_{SG}\right. \\
&\left. - W_{loss,pc}\right)\Big)
\end{aligned}
\tag{3}
$$

IV. ADAPTIVE PID BASED FUZZY LOGIC CONTROLLER DESIGN FOR PRESSURIZED SYSTEM

The PID controller is widely used in engineering control and a variety of sectors. This controller is made up of three terms: proportional (P), integrator (I), and derivative (D). The disparity between the reference set point and the measured signal is fed into the PID controller, and the following mathematical formula determines the output:

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau)\, d\tau + k_d \frac{de(t)}{dt} \tag{4}$$

The error signal $e(t)$ is the difference between the setpoint and the plant output. The PID parameters influence the machine dynamics. The closed-loop step response is assessed in terms of four main characteristics. First, a proportional controller (Kp) is responsible for the rise time reduction and the steady-state error reduction (not elimination). Second, eliminating the steady-state error by an integral controller (Ki) leads to worse transient response. Third, the stability increase, overshoot reduction, and transient response improvement are responsible for a derivative control (K_d). Third, one of the PID controller disadvantages is that the control coefficients (kp, ki, kd) are constant. In other words, throughout the control process, the amount of the impact of the current error, past errors, and error fluctuations in the computation of the current control signal remains constant. Fourth, the PZR system complexity and multivariable conditions make control solutions based on conventional methods (PID) unsatisfactory. To increase control performance, the values of these coefficients may need to be changed over time. In power systems, these typical methods' robustness and reliability allow fuzzy controllers to resolve a wide variety of control problems. Fuzzy logic is among the most often used

applications of a fuzzy set of linguistic instead of numerical variables. Creating a useful approach for creating such control systems calls for combining several artificial intelligence ideas. A fuzzy logic controller (FLC) could be used in place of a PID controller when the given system models are either inaccessible or imperfect.

One method applied for this is the adaptive PID (APID)-based fuzzy logic controller. Different coefficient alternatives should be taken into account in this system since the PID controller coefficients change with time depending on the conditions. Using error and error derivative data (as fuzzy system inputs), the fuzzy logic controller (FLC) can alter the values of the coefficients kp,ki, and kd at any instant. Consequently, the system output of APID controllers follows the reference signal at a fair rate—better than that of the traditional PID controller.

A Mamdani-type fuzzy system is used to change PID coefficients. Mamdani fuzzy systems can provide compensators based on an intuitive user understanding of using a device. For these types of controllers, there is no need of a mathematical model of the machine. Rather, fuzzy controls are mainly constructed with professional knowledge. A Mamdani-type Fuzzy controller consists, as Fig. (4) in four main components: a fuzzifier, an inference system, an information base, and a defuzzifier.

The fuzziness task involves converting the computed quantities of the process (pressure (error, error) and degree (error, error)) into fuzzy sets suitable for the inference system, which delineates the link between the input and output of the developed Fuzzy Logic Controller (FLC). The knowledge base consists of a set of vague "if-then" guidelines clarifying how a professional might handle a tool. The inference process, the brain of the fuzzy control system, and heart simulate expert decision-making by an approximative inference as long as the intended control policy is followed [27]. Generally speaking, the inference process consists in two straightforward tasks: first, determine whether each fuzzy law (from the knowledge base) fits the present input conditions; this is known as "matching". Second, derive conclusions (reaching the suitable control signal) from the knowledge-inputs bases and facts. This is known as "inference."

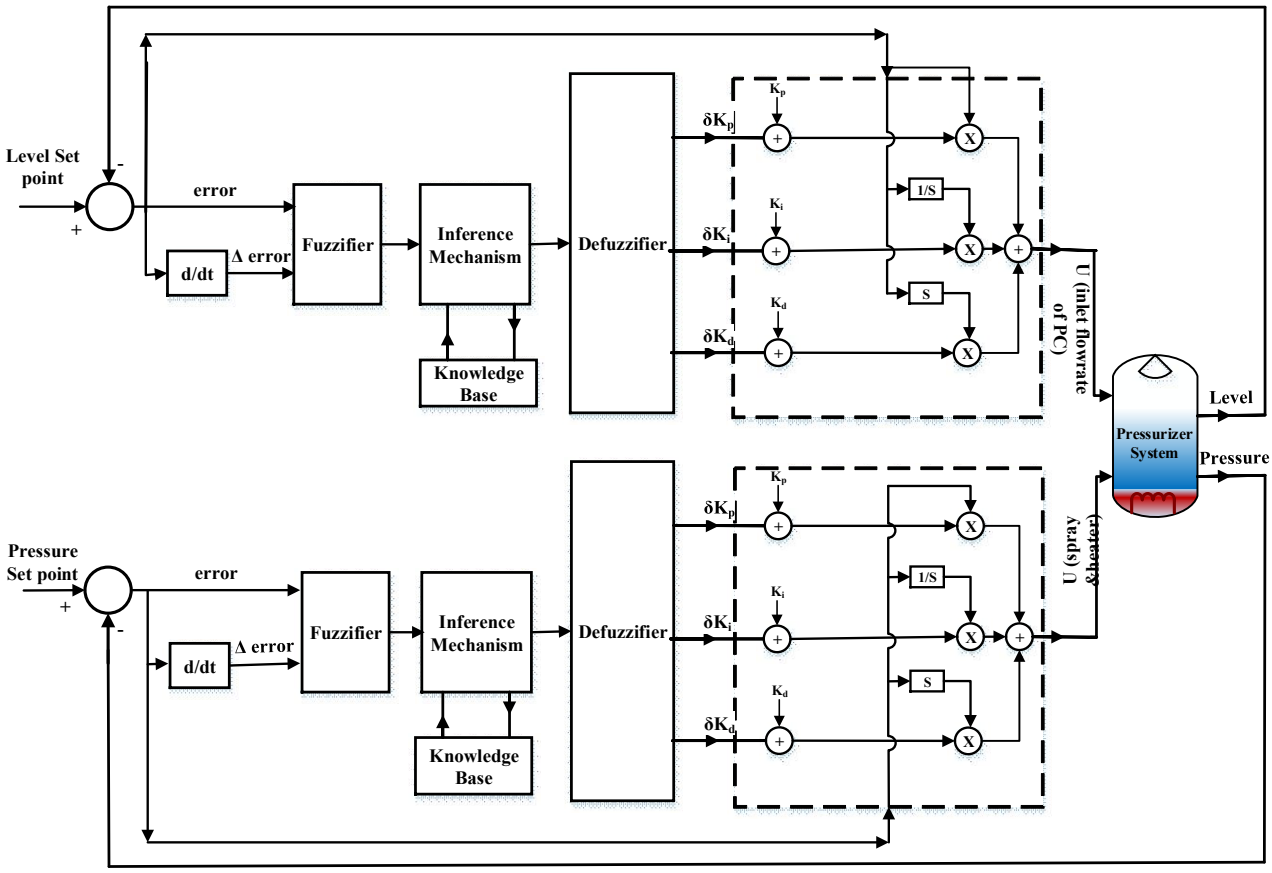


Fig. 4. Block diagram of the Adaptive PID (APID) steps.

The last stage of the FLC architecture—the inference function output (fuzzy output)—is transformed at the defuzzifier into a control action (converted to a crisp number the actuator may apply to the dynamical system). In other words, the purpose of the defuzzification stage is to convert the inference mechanism's fuzzy results into crisp, relevant outputs [28]. The centre of gravity (COG) and centre average (CA) defuzzification are two of the most commonly used approaches for this conversion. The COG method is utilised to defuzzify in this article. The PID control system with a fuzzy gain scheduler is shown in Fig. 4; the approach is to alter the controller coefficients using fuzzy rules and reasoning [28]. The operational response of the PID controller specifies the necessary proportional gain (kp), derivative gain (kd), and integral gain (ki) [28].

FLC is based on a series of fuzzy control rules for linguistic variables of the form:

$$Rule_i: if\ E\ is\ A_i, and\ CE\ is\ B_i\ THEN\ \delta kd\ is\ C_i, and\ \delta ki\ is\ D_i\ and\ is\ E_i \quad (5)$$

$$i = 1,2,\ldots\ldots,N$$

Ai, Bi, Ci, Di and Ei are fuzzy sets on the corresponding supporting sets in these rules. Table (2) illustrates the fuzzy rules for adjusting the proportional gain (δkp). In addition, table (3) and table (4) illustrate the tuning rules for (δki), and (δkd) respectively. The S3, S2, S1, M, P1, P2, and P3 are the names assigned for each membership function as S, M, P represent small, medium, big respectively.

According to the construction and the manner, fuzzy logic to PID control design can be classified into two major categories. First, a standard FLC is built first as a series of heuristic control rules, and then the control signal is deducted directly from the knowledge base. Second, the traditional PID controller gains are tuned in terms of the knowledge base and fuzzy inference, and the APID controller will then produce the control signal.

Table (2): Fuzzy rules of the (δkp).

CE / E	S3	S2	S1	M	P1	P2	P3
S3	P3	P3	P2	P2	P1	M	M
S2	P3	P3	P2	P1	P1	M	S1
S1	P2	P2	P2	P1	M	S1	S1
M	P2	P2	P1	M	S1	S2	S2
P1	P1	P1	M	S1	S1	S2	S2
P2	P1	M	S1	S2	S2	S2	S3
P3	M	M	S2	S2	S2	S3	S3

Table (3): Fuzzy rules of the (δki).

CE / E	S3	S2	S1	M	P1	P2	P3
S3	S3	S3	S2	S2	S1	M	M
S2	S3	S3	S2	S1	S1	M	M
S1	S3	S2	S1	S1	M	P1	P1
M	S2	S2	S1	M	P1	P2	P2
P1	S2	S1	M	P1	P1	P2	P3
P2	M	M	P1	P1	P2	P3	P3
P3	M	M	P1	P2	P2	P3	P3

Table (4): Fuzzy rules of the (δkd).

CE / E	S3	S2	S1	M	P1	P2	P3
S3	P1	S1	S3	S3	S3	S2	P1
S2	P1	S1	S2	S2	S2	S1	M
S1	M	S1	S2	S2	S1	S1	M
M	M	S1	S1	S1	M	S1	M
P1	M	M	M	M	M	M	M
P2	P3	P1	P1	P1	P1	P1	P3
P3	P3	P2	P2	P2	P1	P1	P3

The proposed APID controller is designed in two-levels controller. The fuzzy network is the first level, while the PID controller is the second one. The gains of a PID controller in the APID controller structure are tuned for each control area. The controller gains (δk_p, δk_i, δk_d) are estimated in the first stage based on both the error and the change of error. The estimated gains are then sent to the PID controller (second stage) to perform the following equation operations to calculate K_p, K_i, and K_d new value.

$$K_p = k_p + \delta k_p, \quad K_i = k_i + \delta k_i, \quad K_d = k_d + \delta k_d \qquad (6)$$

Where: K_p, K_i, K_d are the output values after tuning, while k_p, k_i, k_d are the starting values of the PID controller. Also, the output values of the fuzzy controller are (δkp, δki, δkd). Theoretically, it is possible to apply different control devices to the PZR system, but not necessarily each class can be used in practice. As a result, in some sectors, such as PZR, software can be developed from a model that has been evaluated and approved by simulation, and there are different processes between simulation and hardware implementation. The V-cycle development process is used for embedded applications [18, 19, 29], and three tests are often used in these areas: MIL, SIL, and PIL tests. In the next part, the proposed APID controller will be practically applied through the processor in the loop work to ensure the validity and possibility of using it with the PZR

system.

V. PROCESSOR IN THE LOOP IMPLEMENTATION FOR THE PROPOSED SCHEME

The proposed controller with a fixed step at sample time (0.5 sec) is modelled and connected to the PZR model. The controller model is tested in a simulation environment (Simulink) in this step, which is known as the MIL test. MIL denotes that the controller and plant models are simulated on the host computer without the use of any actual hardware components [30-32]. The SIL is created in the second step, where the software is generated from the host computer model. Finally, the third step is concerned with the PIL generation, as presented in Fig. (5).

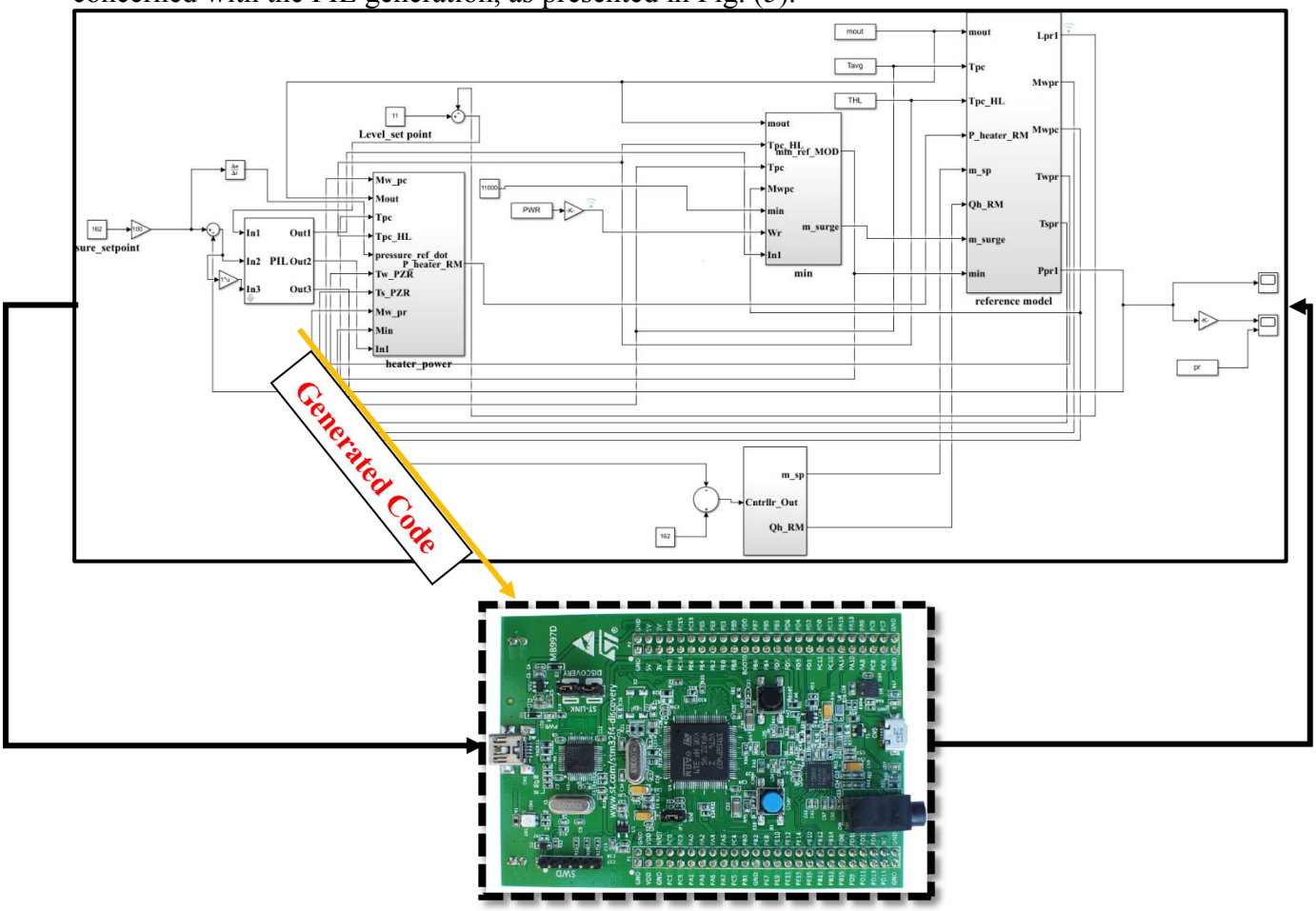


Fig. 5. STM32F407 discovery kit connected to the Simulink model via USB cable.

Once the proposed control algorithm's model is validated using the MIL method, the SIL method automatically checks the code from the proposed control model. In a SIL test, the code for the 64-bit goal is created from the model by using Microsoft Windows SDK (Software Development Kit) and embedded coder tools. The generated code is subsequently validated on the host computer without the use of any additional hardware [30]. The alternative "Create Software-in-the-Loop (SIL) block" must be chosen before constructing the fixed stage controller model to produce the SIL block. Simulink's S-function block is used to execute the created code, and this block is linked to the PZR model. Results of this test are compared with those of the MIL test. Furthermore as you can see, the SIL test follows the MIL outcome. Following the validation of the fixed stage controller algorithm using the SIL approach, the PIL method examines the dynamically produced embedded software and assesses the proposed

controller's implementation. The code is generated from the model for the embedded target, and the generated hex file is loaded and run on the embedded board using the target's compiler and embedded coder function. On the other hand, the host computer simulates the plant model. They also are connected via a USB wire. The PIL test is a fundamental part of the production process since it guarantees that the implementation code satisfies the criteria of our approach [33]. As Figure 5 shows, the embedded board employed in this work is the STM32F4 Discovery board from ST Microelectronics. The microcontroller integrated into this board is STM32F407VG from ST Microelectronics. This microcontroller is equipped with a 32-bit ARM Cortex-M4F CPU and a 1Mbyte flash memory.

VI. RESULTS AND DISCUSSION

This part uses the MATLAB/Simulink software to create a PZR model with a varied set of input variables. The Simulink fixed-step ode45 solver is used to solve the PZR's nonlinear two-region model. Besides, the proposed APID controller's performance considers the load power variation of the PWR. The applied controllers are designed based on the nonlinear model, which describes the dynamic characteristics of the PZR system. In this section, the proposed APID control will be applied to three scenarios: the regular operation of the reactor, loss of reactor coolant flow rate, and setpoint change of PZR pressure. The output behaviour is analyzed and validated based on input data given in table (5). The Matlab Toolbox of PID Tuner is used to tune the PID controllers to control both level and pressure. Hence, several gains for proportional, integral, and derivative are obtained. These parameters are then applied to the PID controllers, and the acquired step responses of each group are recorded.

Table (5): **Reactor (VVER-1200 PWR) [10, 34, 35].**

Symbol	***Value***	***Unit***	***Symbol***	***Value***	***Unit***
Wr	11.654×10^{8}	W	Δ	15	°C
M_{pc}	84000	Kg	T_w	326	◦C
T_{pc}	281.13	°C	T_s	326	◦C
m_{in}	14.222	kg/s	P_{heat}	2.52	kW
m_{out}	4324	kg/s	l_{PZR}	11	M
$T_{pc,hl}$	296.13	°C	p_{PZR}	162	Mpa
$T_{pc,cl}$	266.13	°C	$c_{p,pc}$	6873.1	J/kg/K
$T_{pc,i}$	258.85	°C	W_{loss}	1.6823×10^{5}	W
$c_{p,pc}$	5355	J/kg/K	M_w	19400	Kg
$k_{t,sg}$	9.5296×10^{6}	W/K	M_s	10000	Kg
$W_{loss,pc}$	2.996×10^{7}	W	A_{PZR}	4.52	m^2
$V_{pc,0}$	343	m^3	$V_{PZR,vessel}$	44	m^3
$M_{pc,0}$	0.84×10^{5}	Kg	$p_{*,T}(T)$		par
q_{h}^{max}	0.72	KW	a_0	28884.78	kPa
Back up heater	1.8	KW	a_1	258.01	kPa/°C
W_{sp}^{max}	37	$kg\cdot s^{-1}$	a_2	0.63455	$kPa/°C^2$
c_1	2.98	$kg/m^3/°C$	$\mathcal{P}$ (T)		kg/m^3
c_2	0.00848	$kg/m^3/°C^2$	c_0	581.2	kg/m^3

A. Regular operation with 100% load power

The PID and proposed APID controllers are firstly applied on full load power (100 %), and the obtained response for the water level of PZR is illustrated in Fig. 6. It indicates a damping oscillation around the reference level, which is determined as 11 m. Simultaneously, the obtained pressure response for the different controllers of the PZR is also compared around the setpoint 162 par is illustrated in Fig. 7. Figures (6) to (12) show the acquired system response during these transients. The traditional PID controller, as illustrated in Fig. 7, does not rapidly settle the resulting pressure at its goal value of 162 par, but instead retains a steady-state error for a settling time of 500 sec. It indicates an overshoot of 47.6% and a rise time of 6 sec with constant gains of Kp = 0.28, Ki = 0.01, and Kd = 0.7.

On the other hand, the APID controller application achieves a lower steady-state error with a settling time of 400 sec compared to the conventional ones. The resultant pressure response was achieved by applying the APID controller with the PZR model, as indicated in Fig (7). It illustrates an enhancement of the overshoot to 9.34 % with a rise time of 7.7 sec as listed in table (6). The acquired responses conclude the superiority of using the APID for PZR pressure control, which has a better reaction with appropriate spray and heater actuation, as illustrated in Figs. (10) and (11). Furthermore, these responses are synchronized with the control signals as in Fig (9) with adapted gains of kp1, ki1, kd1 as shown in Fig (12) (b) compared to the conventional PID controller. The obtained PID response to the water level of the PZR system is illustrated in Fig (6). It indicates a damping oscillation around the reference level 11 m. This damping fluctuates between -0.6 % and 0.7%, with a rise time of 3.735 sec as listed in table (7). As shown in Figs (10) and (11), the actuation of spray and heater are synchronized with the control signals in Fig (8) with constant gains Kp2=99990.65, Ki2=89000.004, and Kd2 = 7000.7. The water level's obtained response is also observed for the APID controller, as shown in Fig (6). The resultant overshoot 0.5% and a rise time of 4.5 sec with the control signals as in Fig (8) using adapted gains kp2, ki2, and kd2 in Fig (12) (a). Comparing the two responses indicates the APID controller's superiority as it achieves better performance than the PID controller in each of the overshoot and rise time.

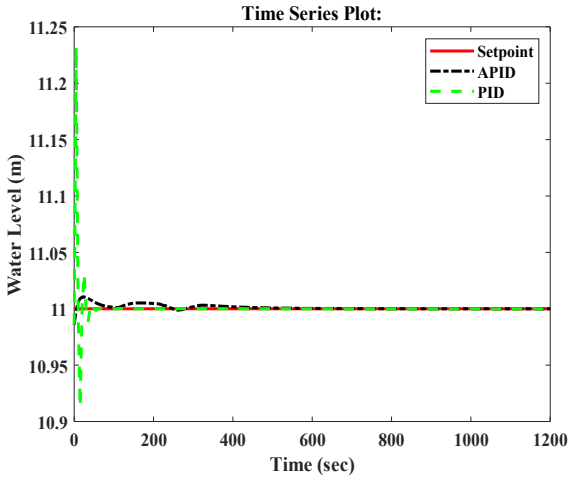


Fig. 6. Water Level Response of PZR for PID and APID Controllers.

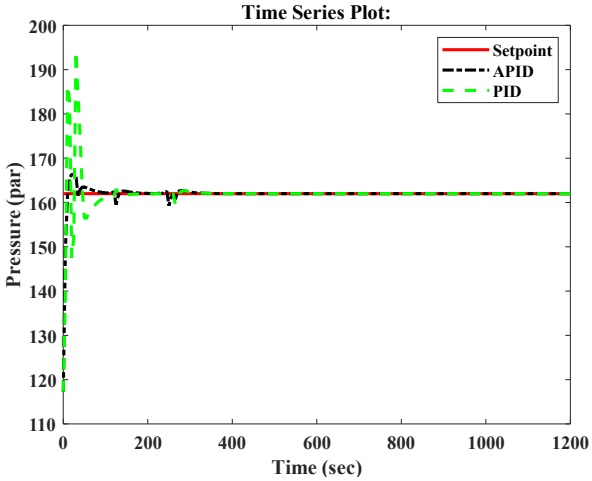


Fig. 7. Pressure Response of PZR System for PID and APID Controllers.

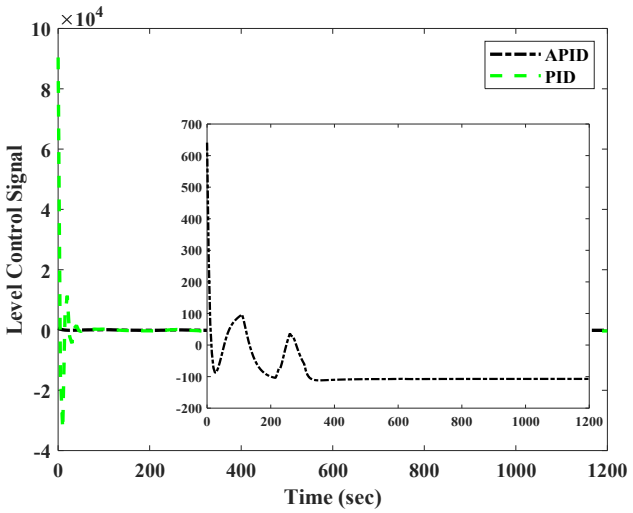


Fig. 8. Water Level control signal for APID Controller.

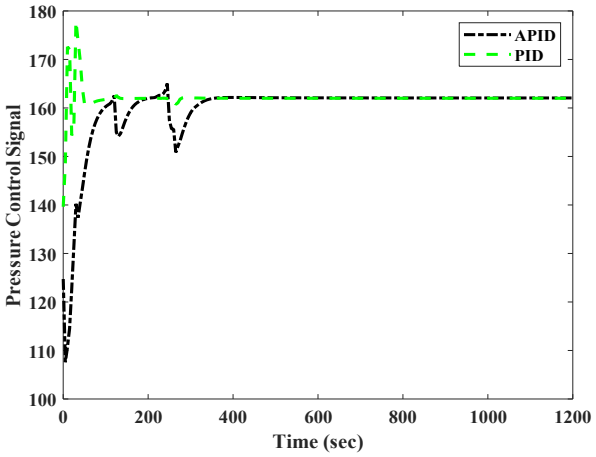


Fig. 9. The pressure control signal for the APID Controller.

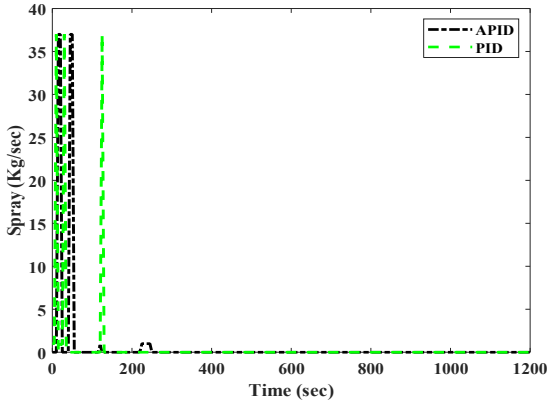


Fig. 10. Spray flow rate for PID and APID controllers.

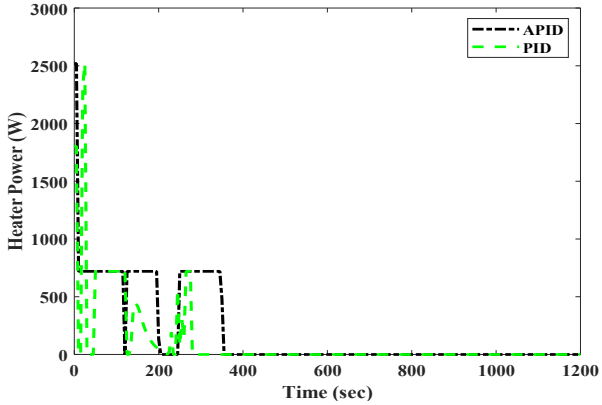
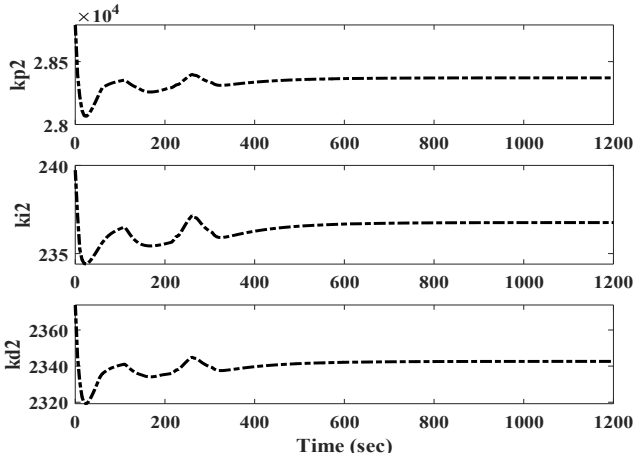
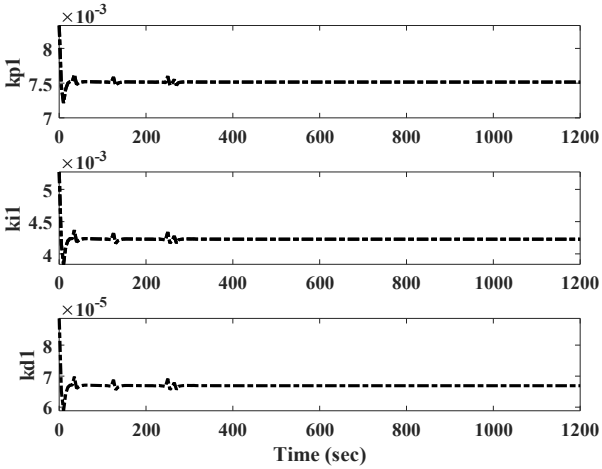


Fig. 11. Heater power for PID and APID Controllers.



(a)



(b)

Fig. 12. Adapted APID gains Controllers.

Table (6): Transient comparison of the PID and APID controller for pressure.

Controller	Rise Time (sec)	Overshoot (%)	Undershoot (%)
PID	6 sec	47.6%	34.4%
APID	7.7 sec	9.3 %	1.38 %

Table (7): Transient comparison of the PID and APID controller for water level.

Controller	Rise Time (sec)	Overshoot (%)	Undershoot (%)
PID	3.735 sec	0.7%	-0.6%
APID	4.5 sec	0.5%	---

B. Normal operation with load power variation

The load power changes between values ranging from 100 %, 80%, 40%, and 60%, up or down within 1200 seconds, as shown in Fig (13). Finally, the proposed APID controller was applied, and the conventional PID controller compared them, as shown in Fig (14), which displays the water level response, and Fig (15) indicates the pressure response.

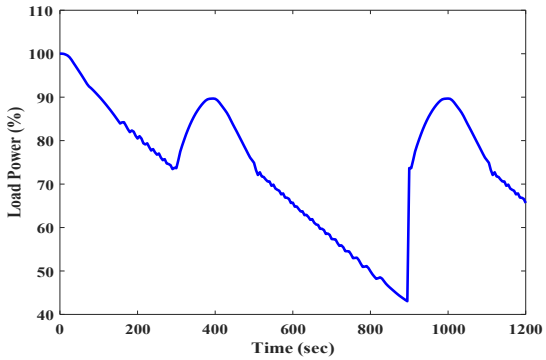


Fig. 13. Load Power Variation.

The obtained PID response to the water level of the PZR system is illustrated in Fig (14). It indicates a damping oscillation around the reference level 11 m with a rise time of 49 sec as listed in the table (9) with an overshoot of 10.6 % and undershoot -0.09 % constant gains K_{p2}=9999.65, K_{i2}= 89000.1, and K_{d2}=7000.7. As shown in Figs (18) and (19), the actuation of the spray and heater are synchronized with water level response in Fig (14) the control signals shown in Fig (16). The water level's obtained response is also observed for the APID controller, as shown in Fig (15) synchronized with the control signals shown in Fig (16) using adapted gains kp2, ki2, and kd2 illustrated in Fig (20) (a). Comparing the two responses indicates the APID controller's superiority as it achieves better performance than the PID controller in each of the overshoot and rise time. The traditional PID controller, as illustrated in Fig. 15, does not rapidly settle the acquired pressure at its target value of 162 par, but instead keeps a steady-state error for a rising time of 5.4 sec. It indicates an overshoot of 86.7 % and undershoots of -86.7 % with constant gains of K_p = 0.38, K_i = 0.01, and K_d = 0.4.

On the other hand, the APID controller application enhances the overshoot to 4.9 % and undershoot 10.5 % with an accepted rise time to the PZR system of 11.5 sec as listed in table (9). The acquired responses confirm the superiority of using the APID for PZR pressure control, with better pressure response and excellent spray and heater actuation, as illustrated in Figs. (18) and (19). Furthermore, these responses are synchronized with the control signals as in Fig (17) with adapted gains of kp1, ki1, kd1 as shown in Fig (20) (b) compared to the conventional PID controller.

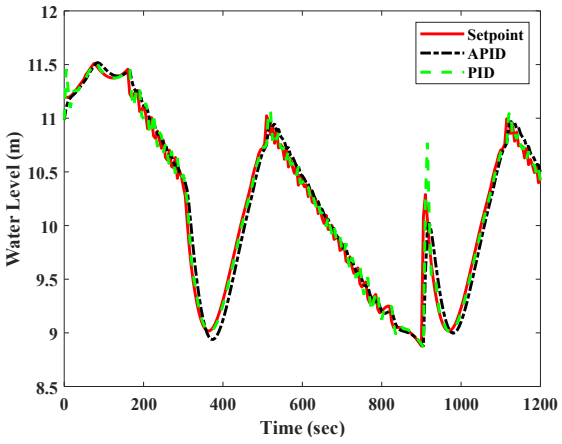


Fig. 14. Water Level Response of PZR for PID and APID Controllers.

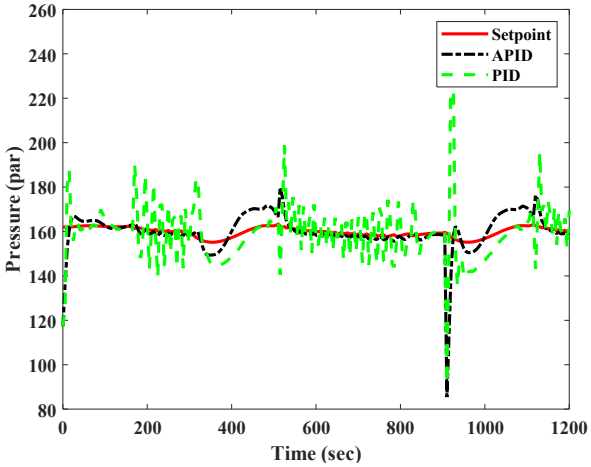


Fig. 15. Pressure Response of PZR System for PID and APID Controllers.

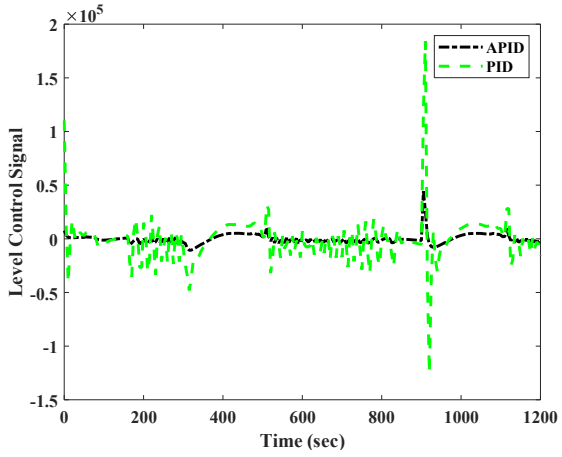


Fig. 16. Water Level control signal for APID Controller.

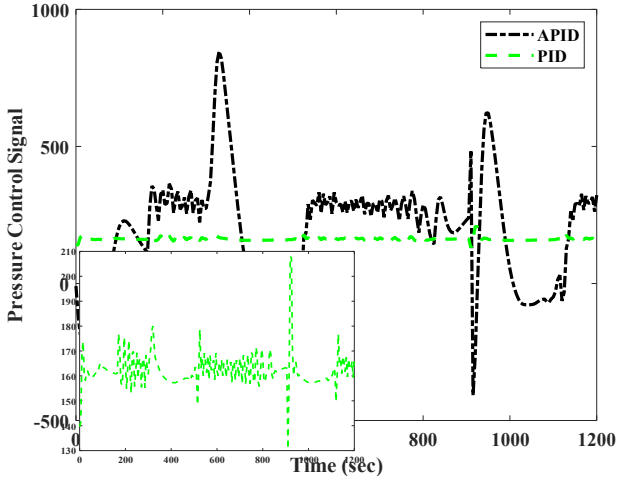


Fig. 17. The pressure control signal for the APID Controller.

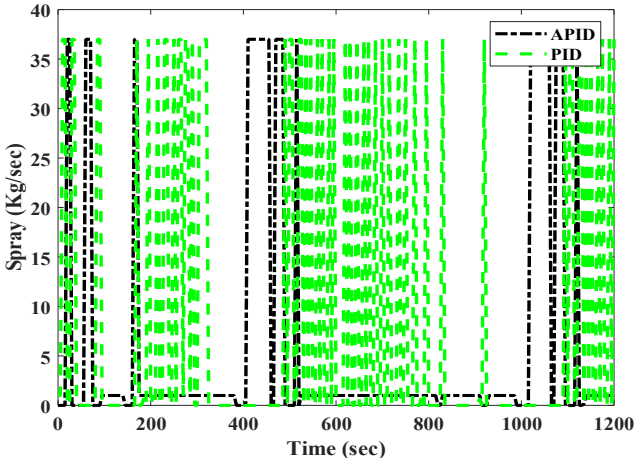


Fig. 18. Spray flow rate for PID and APID controllers.

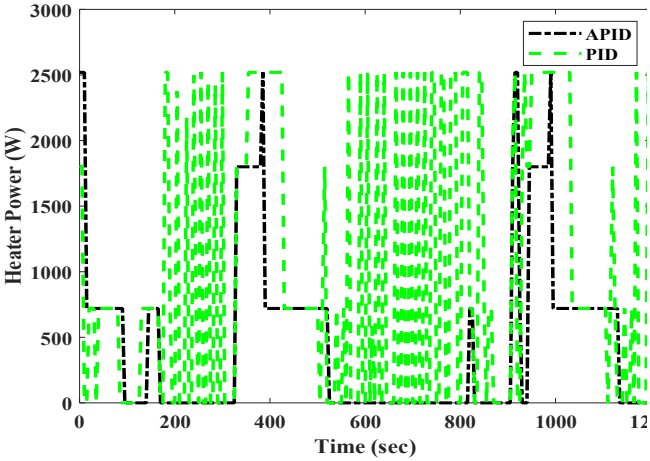
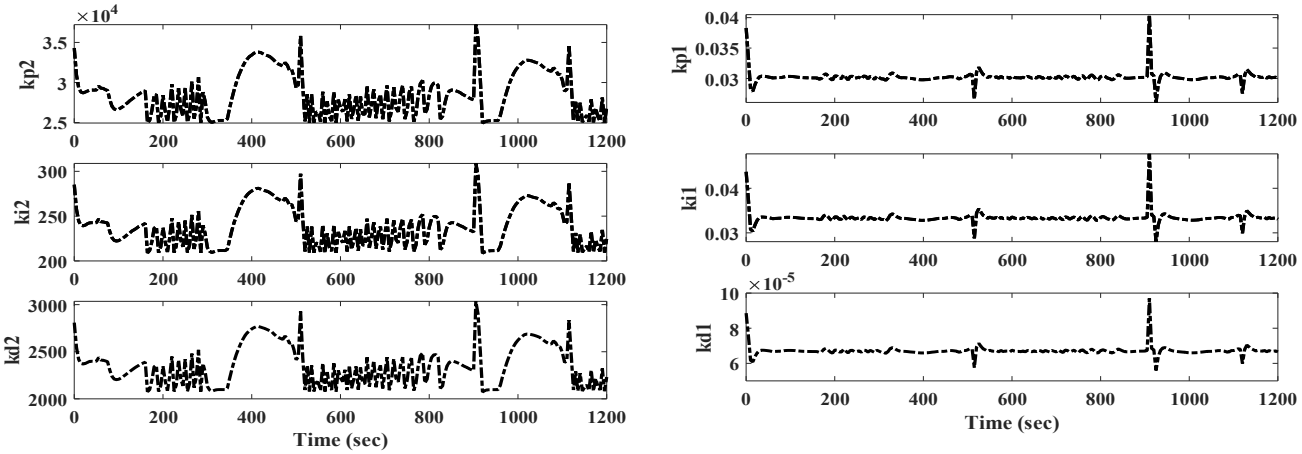


Fig. 19. Heater power for PID and APID Controllers.



(a) (b)

Fig. 20. Adapted APID gains Controllers.

Table (8): Transient comparison of the PID and APID controller for pressure.

Controller	Rise Time (sec)	Overshoot (%)	Undershoot (%)
PID	5.435 sec	86.7 % / 86.7	-86.7 % /16%
APID	11.5 sec	4.9% / -0.6%	10.5% / 0.6%

Table (9): Transient comparison of the PID and APID controller for water level.

Controller	Rise Time (sec)	Overshoot (%)	Undershoot (%)
PID	49 sec	10.6% / 1.5%	-0.09% / 2.3%
APID	---	---	---

C. Loss of Reactor Coolant Inlet Flow

The PWR VVER-1200 simulator is subjected to malfunction loss of reactor coolant inlet flow is described in details in [36, 37].

Reactor coolant pumps (RCPs) give the PC system the pressure head required for the forced circulation of the reactor coolant. Because of the lower coolant flow rate, the trip of one or more RCPs rapidly raises fuel temperature under typical operation conditions.

The reactor output is consequently diminished by approximately 30%, as illustrated in ***Fig. (21***) The reactor protection system autonomously initiates a reactor trip, which is promptly followed by a turbine trip, as the turbine trip invariably succeeds a reactor trip. Following the reactor and turbine shutdowns, a negative flow rate is detected in the RCP, signifying a reversal of flow direction in this loop.

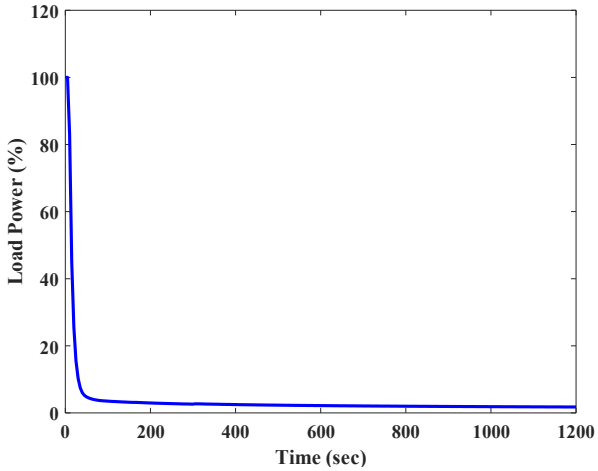


Fig. 21. Change of PWR load power.

When the RCP is manually deactivated, the average PC temperature Tavg increases due to a reduction in coolant circulation within this loop, resulting in diminished coolant flow to dissipate heat from the core. Without control rod action, reactor output often declines by around 30% within the first 10 seconds following the RCP trip.

Less moderation follows from the core's decreased cooling flow. Negative reactivity also comes from the temperature of the negative moderator increasing as the Tavg increases. Because the flow rate of coolant is low, the reactor trip alert is sent in 10 seconds. During the reactor ride, the turbine often trips. Over time, natural circulation builds in this loop, which helps get rid of any leftover heat in the heart.

When the coolant flow is changed in loop-B, the flow rate becomes negative, as shown in Fig. 22, and the cold-leg temperature of this loop is higher than the hot-leg temperature.

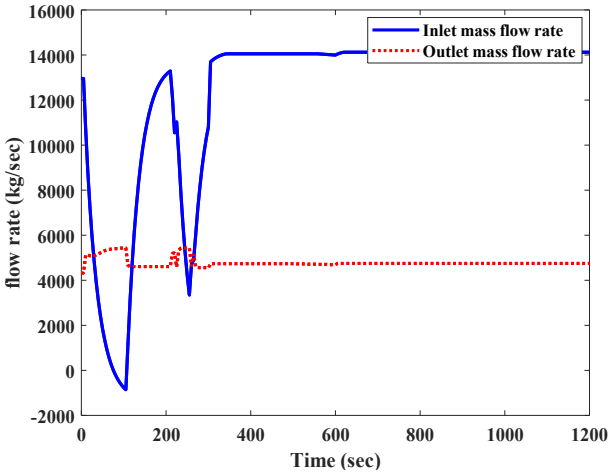


Fig. 22. Inlet flow rate m_{in}, Outlet flow rate m_{out}.

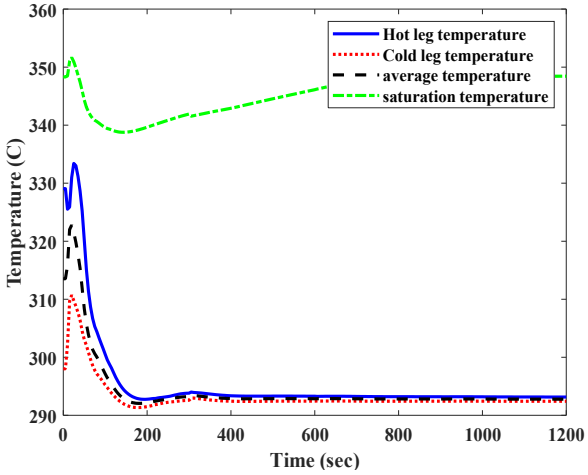


Fig. 23. Cold-leg $T_{PC,CL}$, Hot-leg $T_{PC,HL}$, average T_{PC}, and saturation T_{sat} Temperature.

Figure 23 illustrates the coolant temperatures in the cold-leg and hot-leg of the loop where the RCP was activated. The initial decrease in hot-leg coolant temperature results from diminished core strength, while the reduction in coolant pressure flow is attributed to an increase in cold-leg temperature.

The reactor trip causes the temperature of the loop cold-leg coolant to rapidly drop after around 10 seconds. The heat of oxidation causes the hot-leg temperature to rise somewhat after about 30 seconds; backflow in this loop causes it to rapidly drop after about 40 seconds.

Backflow in this loop causes the cold-leg coolant's temperature to rise beyond the hot-leg coolant's after around 60 seconds. The surge flow rate generated by the swelling and shrinkage of RCS coolant during transients is computed using data obtained by the PCtran VVER-1200 simulator in figure 24. As mentioned in equation (1), the nonlinear PZR model is subject to various APID controller settings.

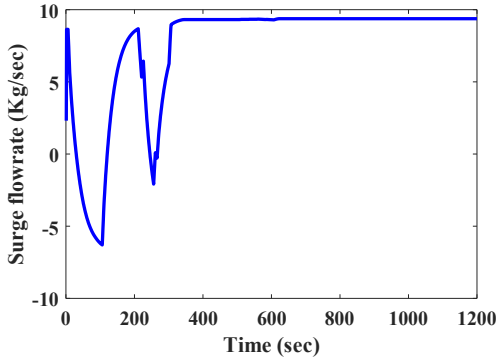


Fig. 24. Surge flowrate of pressurizer system.

Figures (25) to (27) show the obtained system response during these transients. As shown in Fig. 28, the conventional PID controller does not quickly settle the obtained pressure at its target value of 162 par, but instead maintains an overshoot of 97.8 percent and a rise time of 1.83 sec. The actuation of spray and heaters shown in Fig. 29 and Fig. 30 are synchronized with control signals in Fig (28) with constant gains K_p=0.28, K_i=0.1, and K_d=0.7. On the other hand, the APID controller application achieves a lower steady-state error with an overshoot to 9.34 % with a rise time of 4 sec, as listed in table (10). The obtained responses conclude that the APID is superior for controlling the PZR pressure because it has a better pressure response synchronised with the control signals shown in Fig. 28 with adapted gains of kp1, ki1, and kd1 as shown in Fig. (31) (b). The obtained water level PID response of the PZR system is illustrated in Fig. 25 with an overshoot of 54 % and undershoot 34 %, as listed in table (11). Also, the surge flow rate's actuation is synchronized with the control signals as in Fig. 27 with constant gains of K_{p2}=9999.65, K_{i2}=8900.004, and K_{d2}=7000.7. The water level's obtained response is observed for the APID controller, as shown in Fig. 25. The resultant overshoot 1.7 % with the control signals as in Fig. 27 with adapted gains of k_{p2}, k_{i2}, and k_{d2} in Fig. (31) (a). Comparing the two responses indicates the APID controller's superiority as it achieves better performance than the PID controller in each of the overshoot and rise time.

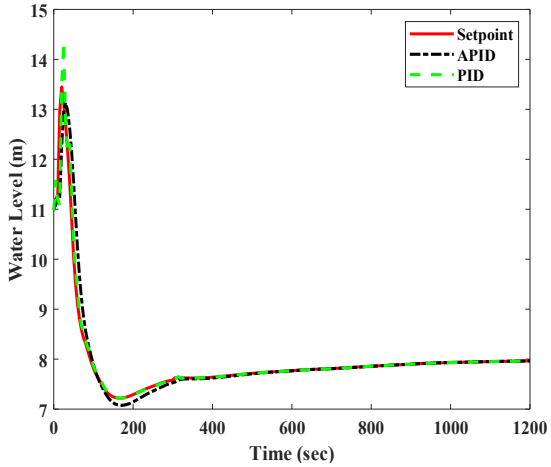


Fig. 25. Water Level Response of PZR for PID and APID Controllers.

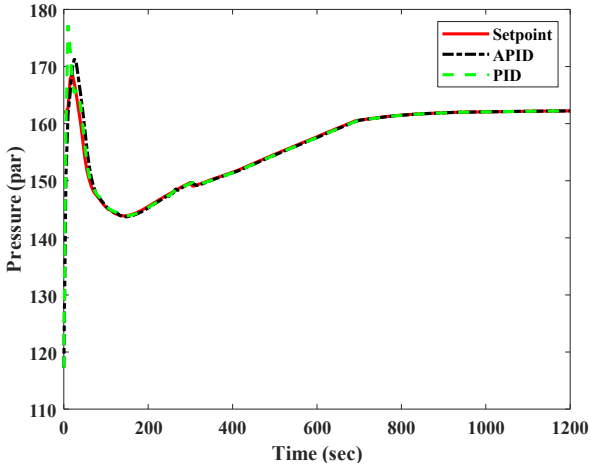


Fig. 26. Pressure Response of PZR System for PID and APID Controllers.

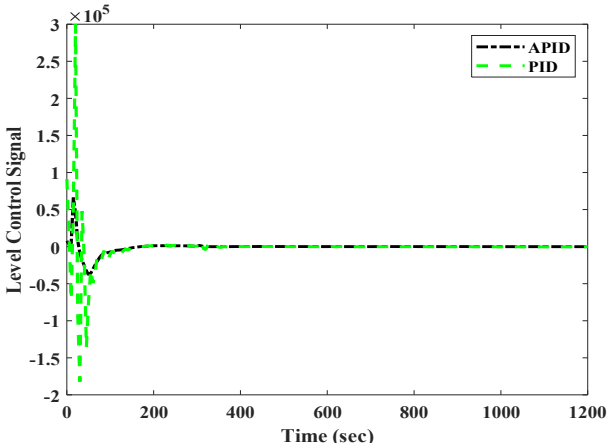


Fig. 27. Water Level control signal for APID Controller.

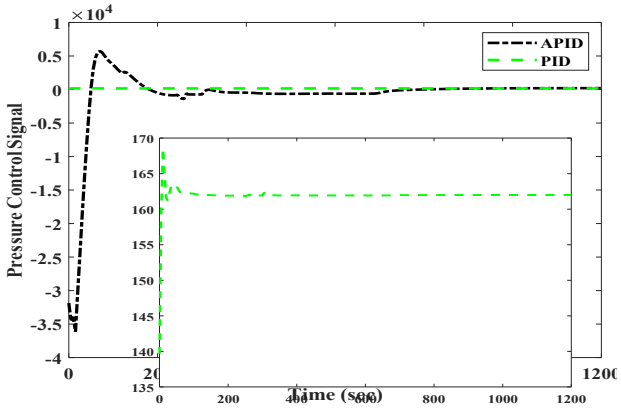


Fig. 28. The pressure control signal for the APID Controller.

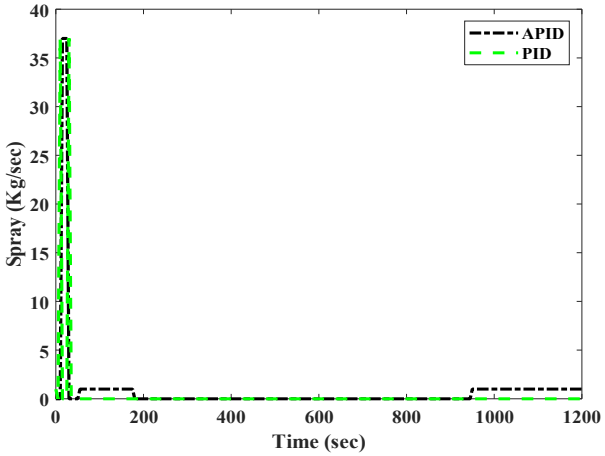


Fig. 29. Spray flow rate for PID and APID controllers.

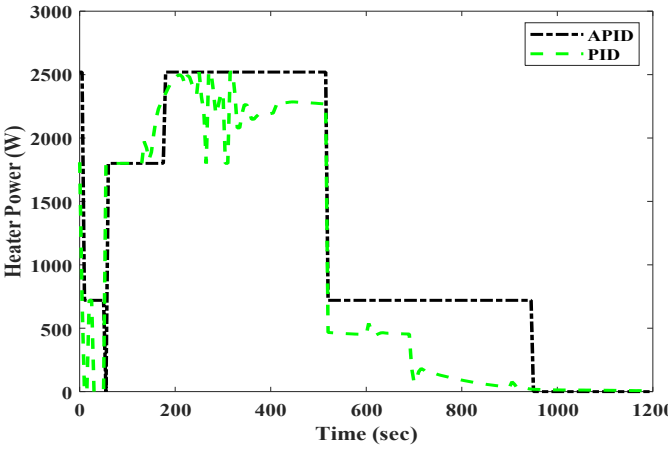
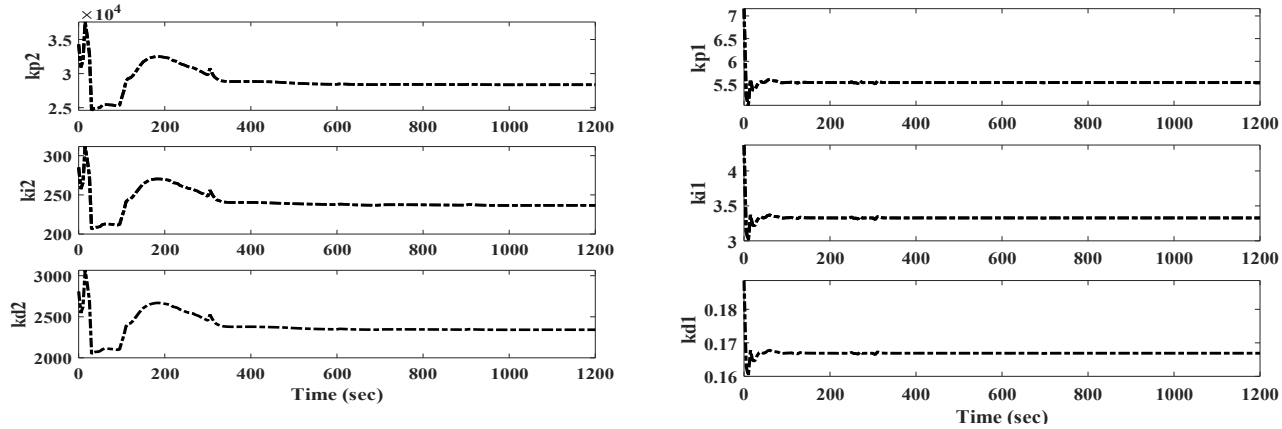


Fig. 30. Heater power for PID and APID Controllers.



(a) (b)

Fig. 31. Adapted APID gains Controllers.

Table (10): Transient comparison of the PID and APID controller for pressure.

Controller	Rise Time (sec)	Overshoot (%)	Undershoot (%)
PID	97.8 %	1.83 %	---
APID	9.3 %	4 sec	---

Table (11): Transient comparison of the PID and APID controller for water level.

Controller	Rise Time (sec)	Overshoot (%)	Undershoot (%)
PID	---	54 %	34 %
APID	---	1.7 %	17 %

D. The first scenario for setpoint change

The operating point has been changed for each pressure from 162 par to 150 par. Thus, the water level changed from 11 m to 8 m inside the PZR to see the APID controller's reaction compared to the PID controller. The obtained system response during these transients is indicated in Fig. 32 to Fig. 38. The accepted PID response to the water level of the PZR system is illustrated in Fig. 32. This figure shows the level change from 11 meters to 8 meters with overshoot -0.7 % and 0.7% undershoot with a rise time of 4 sec as listed in the table (12). Fig. 34 shows the control signal with constant gains of K_{p2}=28330, K_{i2}=234.5, and K_{d2}=2320. Besides the water level obtained response is also observed for the APID controller, the resultant overshoot 0.4% and a rise time of 4.5 sec with the control signals as illustrated in Fig. 34 with adapted gains of kp2, ki2, and kd2 as shown in Fig. (38) (a). Comparing the two responses indicates the APID controller's superiority as it achieves better performance than the PID controller in each overshoot and approximately rises time. As shown in Fig. 33, the conventional PID controller does not quickly settle the obtained pressure at its target value of 162 par but retains a steady-state error for a settling time of 100 sec. As a result, it indicates an overshoot of -0.5 % and a rise time of 2.129 sec. The actuation of spray and heaters shown in Fig. 36 and Fig. 37 are synchronized with the control signal with constant gains of Kp = 0.26, Ki = 0.01, and Kd = 0.09.

On the other hand, the APID controller application achieves a lower steady-state error with settling time than the conventional ones. The resultant pressure response was achieved by applying the APID controller with the PZR model. It illustrates an enhancement of the overshoot to -0.4% with a rise time of 1.8 sec as listed in the table (13). The obtained responses conclude the superiority of using the

APID for PZR pressure control, with better pressure response and appropriate spray and heater actuation, as shown in Fig. 36 and Fig. 37. Also synchronized with the control signals as in Fig. 35 with adapted gains of kp1, ki1, and kd1 as shown in Fig. (38) (b) depending on the lookup table compared to the conventional PID controller.

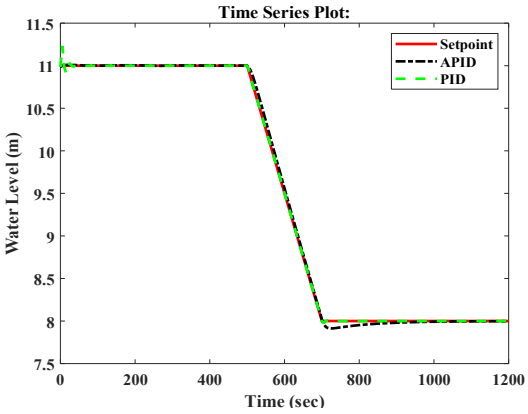


Fig. 32. Water Level Response of PZR for PID and APID Controllers.

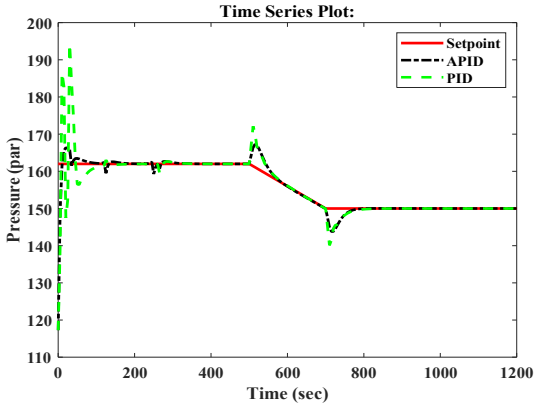


Fig. 33. Pressure Response of PZR System for PID and APID Controllers.

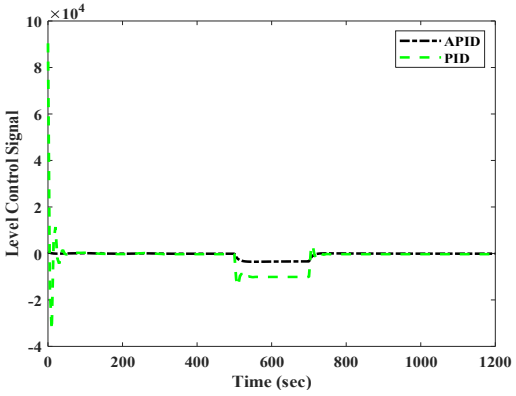


Fig. 34. Water level control signal for PID and APID Controllers.

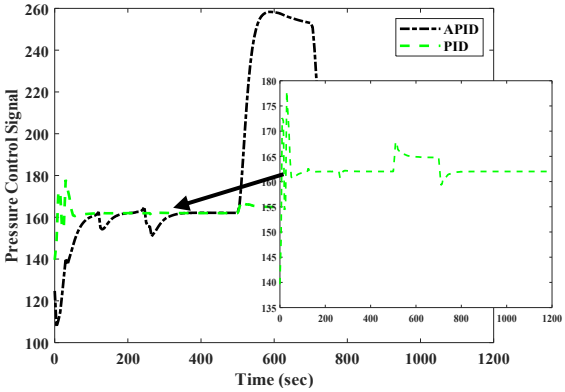


Fig. 35. The pressure control signal for PID Controller.

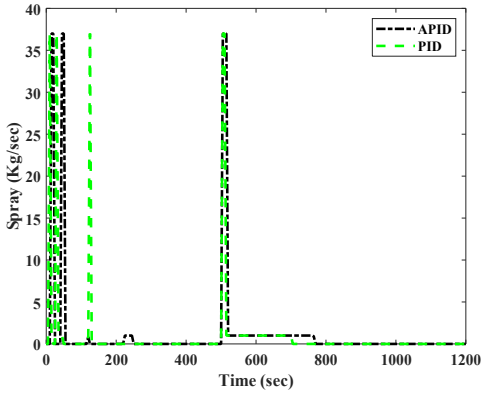


Fig. 36. Spray flow rate for PID and APID controllers.

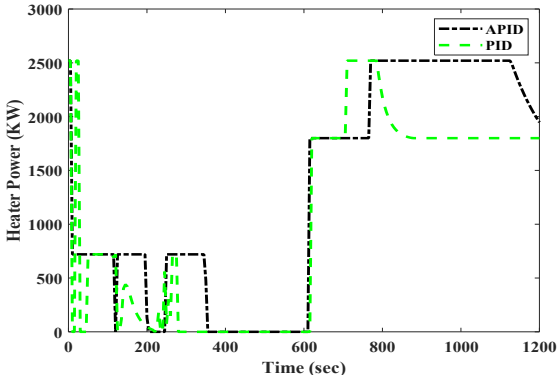
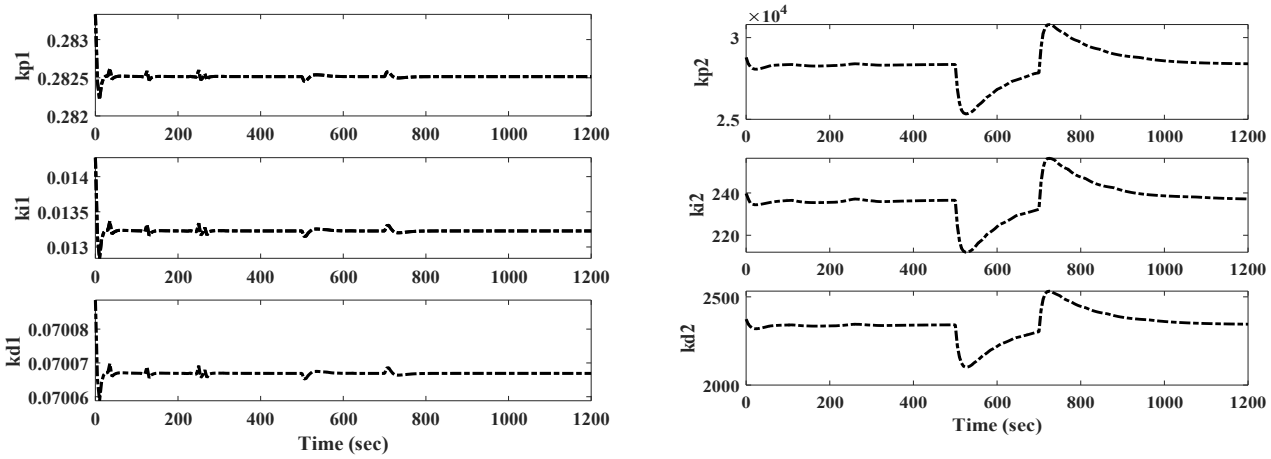


Fig. 37. Heater power for PID and APID Controllers.



(a) (b)

Fig. 38. Adapted gains for Pressure and Level using APID controller.

Table (12): Transient comparison of the PID and APID controller for water level.

Controller	Rise Time (sec)	Overshoot (%)	Undershoot (%)
PID	4 sec	-0.7	0.7
APID	4.4 sec	0.4	0.6

Table (13): Transient comparison of the PID and APID controller for pressure.

Controller	Rise Time (sec)	Overshoot (%)	Undershoot (%)
PID	2.129 sec	-0.5	0.5
APID	1.8 sec	-0.4%	-----

E. The second scenario for setpoint change

The operating point has been changed for each of the pressure from 162 par to 150 par and then to 162 par, and thus the water level changed from 11 m to 8 m then to 11 m inside the PZR to see the APID controller's reaction compared to the PID controller. The obtained system response during these transients is indicated in Fig (39) to Fig (45). The obtained response of PID and proposed APID controllers to the water level of the PZR system are illustrated in Fig (39). This figure shows the change of the level from 11 meters to 8 meters and then back to 11 m with overshoot, undershoot, and a rise in the table (14). Fig (41) shows the control signal with constant gains of K_{p2}=28330, K_{i2}=234.5, and K_{d2}=2320 for PID controller and with adapted gains of kp2, ki2, and kd2 as shown in Fig (45) (a) for APID controller. Comparing the two responses indicates the APID controller's superiority as it achieves better performance than the PID controller in each overshoot and approximately rises time. As shown in Fig (40), the conventional PID controller does not quickly settle the obtained pressure at its target value of 162 par but retains a steady-state error. The rise time is listed in the table (15). The actuation of spray and heaters shown in Fig (43) and Fig (44) are synchronized with the control signals as in Fig (42) with constant gains of K_p= 0.26, K_i=0.01, and K_d=0.09 for PID controller.

On the other hand, the APID controller application achieves a lower steady-state error with settling time than the conventional ones, as shown in Fig (40). The resultant pressure response was achieved by applying the APID controller with the PZR model. It illustrates an enhancement of the overshoot, rise time, as listed in table (15). The obtained responses conclude the superiority of using the APID for PZR

pressure control, with better pressure response and appropriate spray and heater actuation, as shown in Fig. 36 and Fig. 37. The obtained responses conclude that the APID is superior for controlling the PZR pressure because it has a better pressure response with appropriate spray and heater actuation as shown in Figs. (43) and (44) and is synchronised with the control signals as shown in Fig. 42 with adapted gains of kp1, ki1, and kd1 as shown in Fig (45) (b).

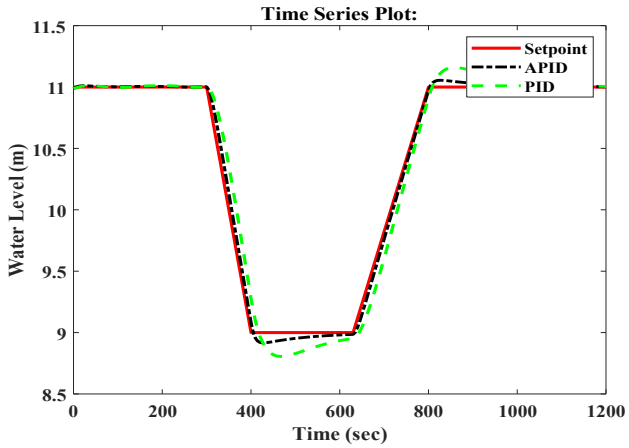


Fig. 39. Water Level Response of PZR for PID and APID Controllers.

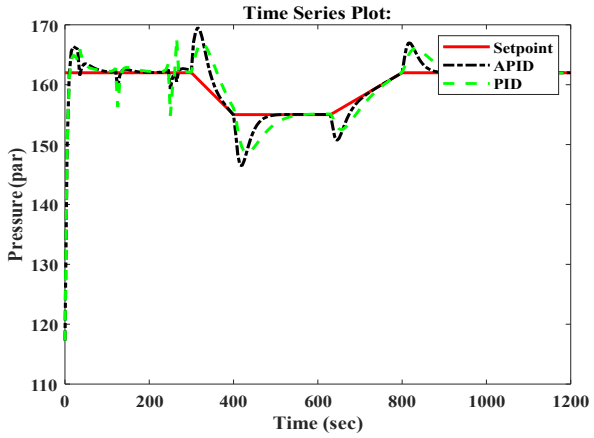


Fig. 40. Pressure Response of PZR System for PID and APID Controllers.

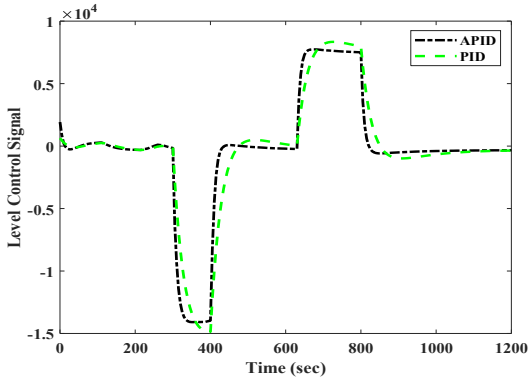


Fig. 41. Water level control signal for PID and APID Controllers.

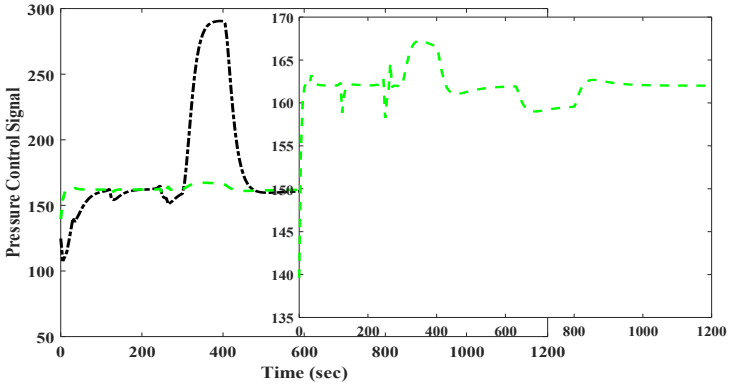


Fig. 42. A pressure control signal for PID Controller.

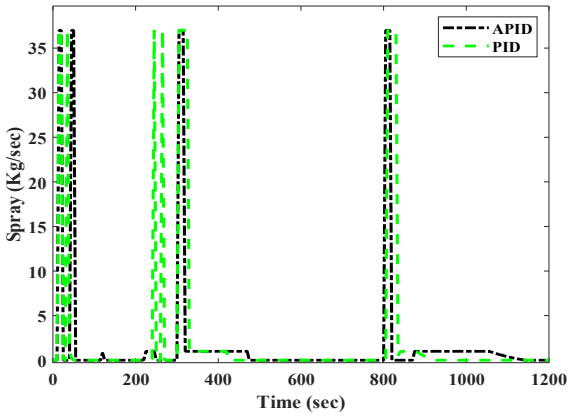


Fig. 43. Spray flow rate for PID and APID controllers.

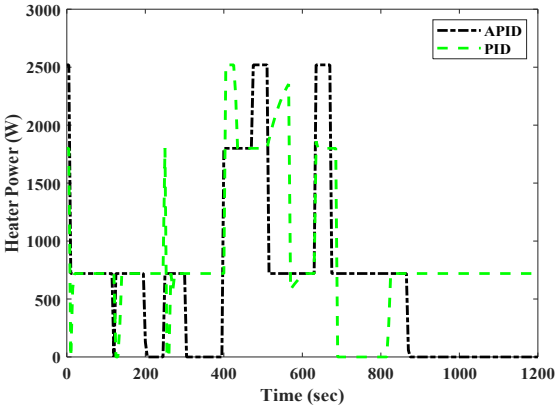
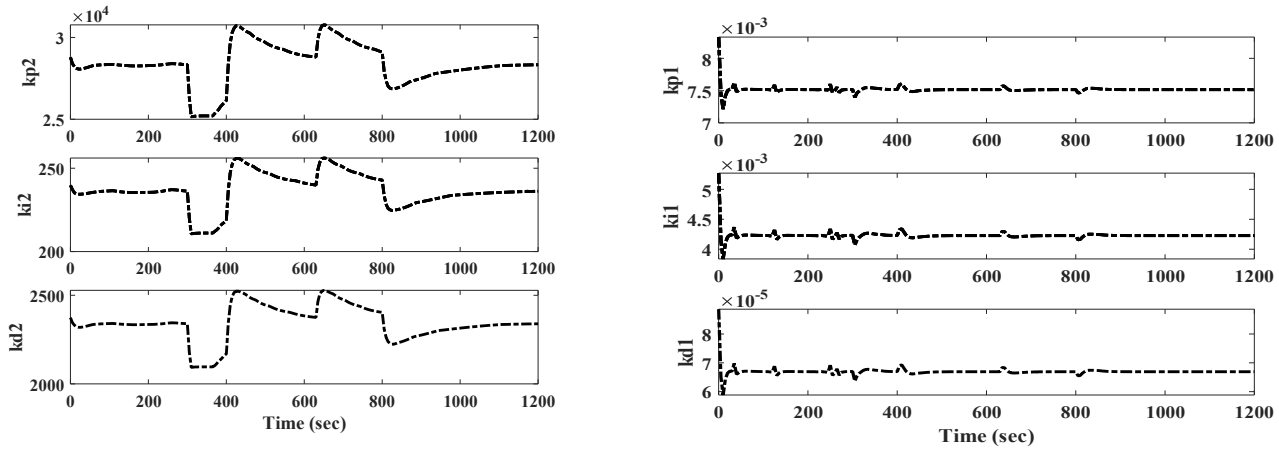


Fig. 44. Heater power for PID and APID Controllers.



(a) (b)

Fig. 45. Adapted gains for Pressure and Level using APID controller.

Table (14): Transient comparison of the PID and APID controller for water level.

Controller	Rise Time (sec)	Overshoot (%)
PID	1.715	5.7%
APID	1.9 sec	5.5%

Table (15): Transient comparison of the PID and APID controller for pressure.

Controller	Rise Time (sec)	Overshoot (%)
PID	682.27 m sec	7%
APID	22.8 m sec	1 %

A. Implementation of Adaptive PID on STM32F407 discovery kit

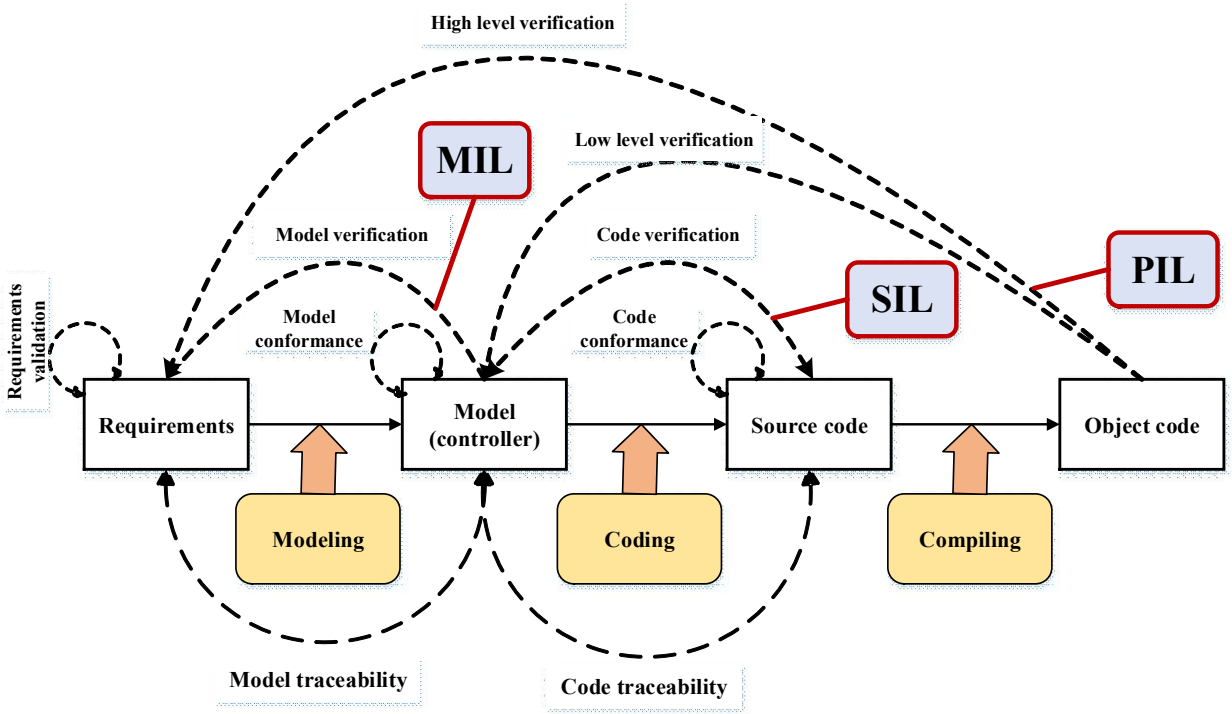


Fig. 46. **MIL/ SIL/ PIL verification steps.**

The embedded target code is created from the model using the target's compiler and embedded coder function, which allows the generated hex file to be loaded and run on the embedded board. The host computer, on the other hand, simulates a plant. They are likewise connected with a USB cable. As shown in Fig. 46, the PIL test process is a crucial part in the development cycle for providing that the implementation code corresponds to the proposed controller's specifications. The STM32F407 discovery kit is used to implement PID and APID controller for all the above scenarios for the PZR system under the sample time of 0.5 sec. The obtained water level and pressure responses are shown from Fig (47) to Fig (56). The obtained performance of the implemented controller, as observed, is similar to approximately the exact behaviour of a simulated controller response.

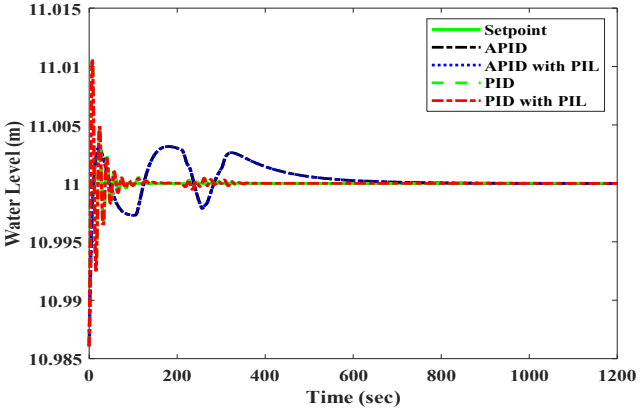


Fig. 47. Implementation of APID Water level Response of PZR System.

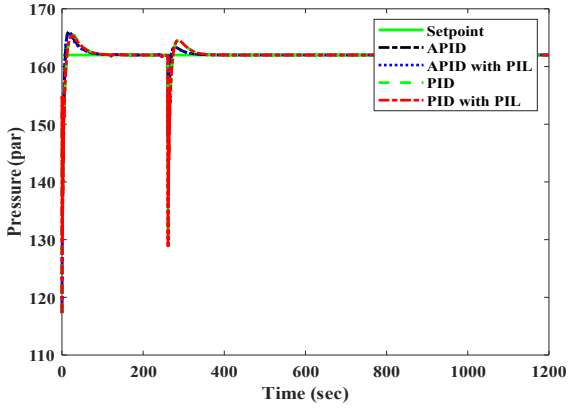


Fig. 48. Implementation of APID Pressure Response of PZR System.

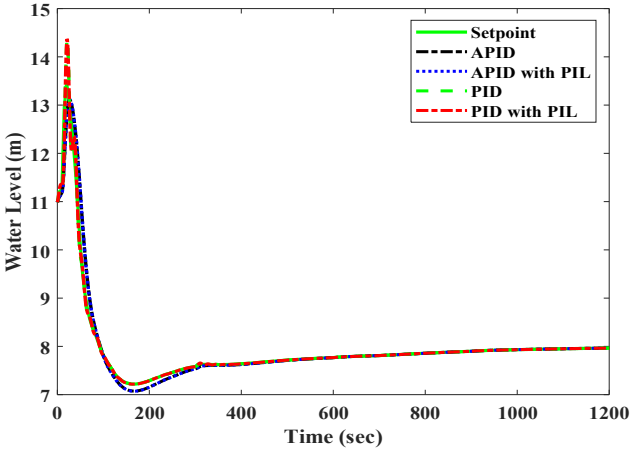


Fig. 49. Implementation of APID Water level Response of PZR System.

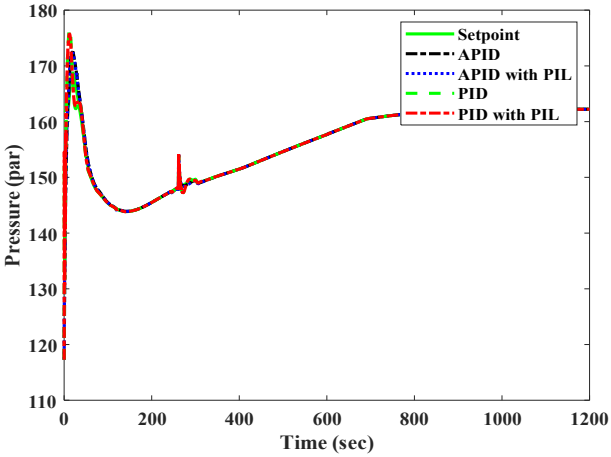


Fig. 50. Implementation of APID Pressure Response of PZR System.

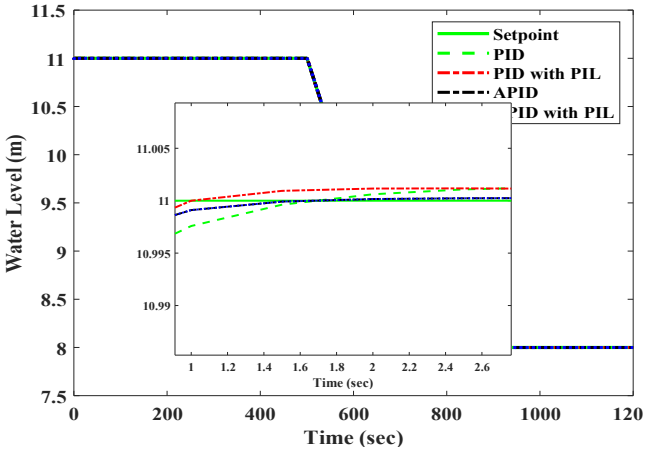


Fig. 51. Implementation of APID Water level Response of PZR System.

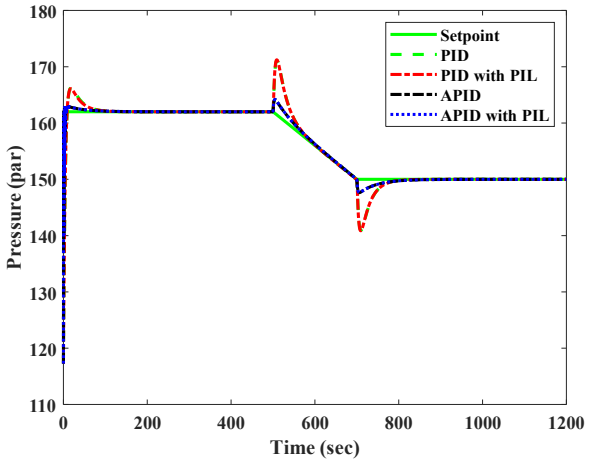


Fig. 52. Implementation of APID Pressure Response of PZR System.

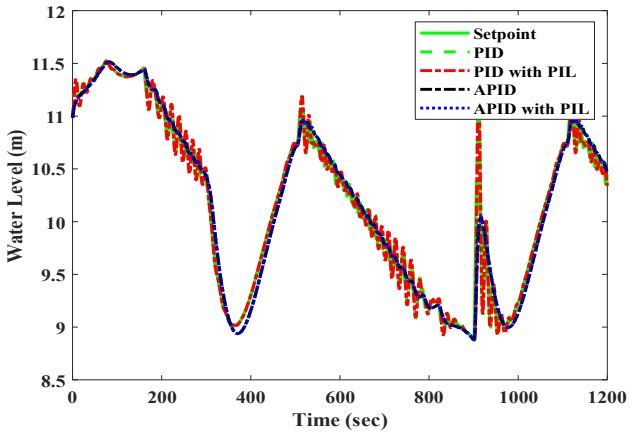


Fig. 53. Implementation of APID Water level Response of PZR System.

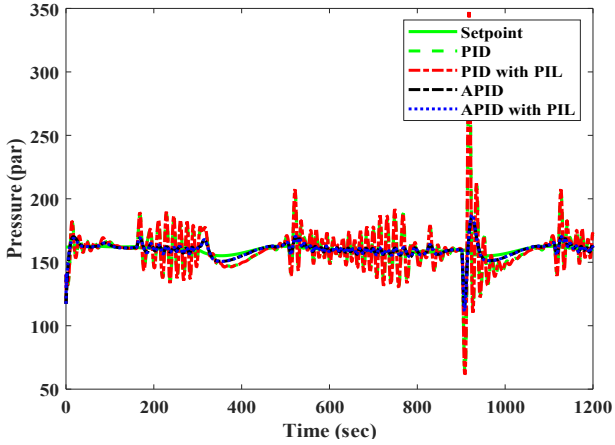


Fig. 54. Implementation of APID Pressure Response of PZR System.

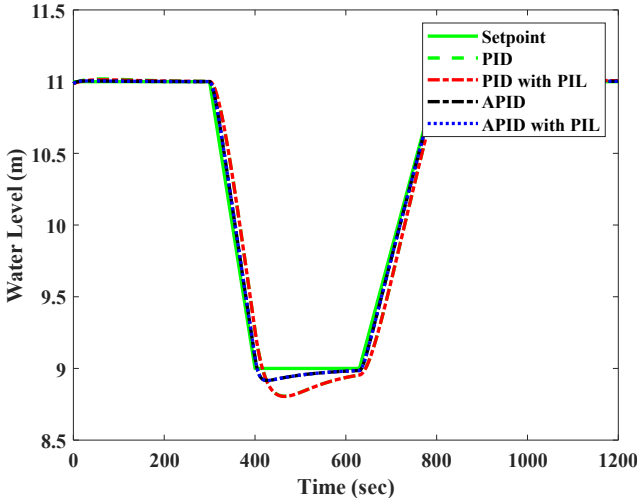


Fig. 55. Implementation of APID Water level Response of PZR System.

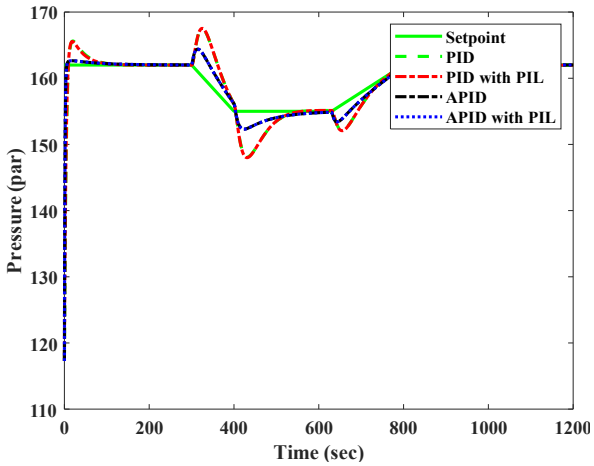


Fig. 56. Implementation of APID Pressure Response of PZR System.

The PZR system response to pressure and water level is shown in the figures under the different control systems of PID, PID with Processor-In-the-Loop (PIL), Adaptive PID (APID), and APID with PIL. Compared to traditional PID controllers, which show observable oscillations and longer response times, Fig. (55), illustrates how well APID-based controllers especially APID with PIL perform in managing the water level with less overshoot, smoother transitions, and faster recovery. The pressure response is shown in Fig. (56), where the APID with PIL technique again performs better than PID controllers by lowering pressure fluctuations greatly and providing a faster and more stable return to the setpoint. The APID systems have a significantly more controlled and stable response than the PID with PIL controller, which displays noticeable instability with significant overshoot and delayed recovery.

VII. CONCLUSION

This work provided an extensive analysis of the design and implementation of an adaptive PID-based fuzzy logic controller for the pressurizer system in Pressurized Water Reactors. The proposed control system was designed to regulate both pressure and water levels, tackling the nonlinear and multi-input multi-output (MIMO) characteristics of the pressurizer. The system got validation using a Processor-in-the-Loop (PIL) test on the STM32F407 discovery kit, demonstrating its practical usefulness. The proposed controller and PIL implementation were validated using a non-equilibrium two-region with critical thermodynamic processes of the PZR model. The PZR pressure was controlled by the spray valve operation and electric heaters through increasing or decreasing the pressure respectively. The water level was controlled through the inlet flow rate of the primary circuit. Derivation of water level and pressure control signals for PZR are also introduced. PID and adaptive PID-based fuzzy logic controllers for controlling pressure and water level of PZR are applied through constant setpoint and setpoint change. The processor in the loop is then used for pressure and level control of the PZR system. The obtained implementation and simulation results indicate the adaptive PID controller efficiency in improving the water level response and a PZR system pressure.

REFERENCES

[1] S. E. Arda and K. E. Holbert, "Implementing a pressurized water reactor nuclear power plant model into grid simulations," presented at the 2014 IEEE PES General Meeting | Conference & Exposition, 2014/07, 2014. [Online]. Available: http://dx.doi.org/10.1109/pesgm.2014.6939303.

[2] S.-N. Kim and P. Griffith, "PWR pressurizer modeling," *Nuclear Engineering and Design,* vol. 102, no. 2, pp. 199-209, 1987/06 1987, doi: 10.1016/0029-5493(87)90253-6.

[3] G. Baghban, M. Shayesteh, M. Bahonar, and R. Sayareh, "Simulating and evaluating the pressurizer dynamic behavior in various sizes," *Progress in Nuclear Energy,* vol. 93, pp. 406-417, 2016/11 2016, doi: 10.1016/j.pnucene.2016.09.014.

[4] S. M. H. Mousakazemi, "Control of a PWR nuclear reactor core power using scheduled PID controller with GA, based on two-point kinetics model and adaptive disturbance rejection system," *Annals of Nuclear Energy,* vol. 129, pp. 487-502, 2019/07/01/ 2019, doi: https://doi.org/10.1016/j.anucene.2019.02.019.

[5] G.-D. Zhang, X.-H. Yang, X.-L. Ye, H. Xu, D.-Q. Lu, and W. Chen, "Research on Pressurizer Water Level Control of Pressurized Water Reactor Nuclear Power Station," *Energy Procedia,* vol. 16, pp. 849-855, 2012, doi: 10.1016/j.egypro.2012.01.136.

[6] G. D. Zhang, X. H. Yang, S. Y. Qiao, and Y. J. Wu, "Research on Pressurizer Pressure Control System of 900MW Pressurized Water Reactor Nuclear Power Plant," *Advanced Materials*

Research, vol. 718-720, pp. 1215-1220, 2013/07 2013, doi: 10.4028/www.scientific.net/amr.718-720.1215.

[7] M. Lotfi, M. B. Menhaj, S. A. Hosseini, and A. S. Shirani, "A design of switching supervisory control based on fuzzy-PID controllers for VVER-1000 pressurizer system with RELAP5 and MATLAB coupling," *Annals of Nuclear Energy,* vol. 147, p. 107625, 2020/11 2020, doi: 10.1016/j.anucene.2020.107625.

[8] P. Wang, J. He, X. Wei, and F. Zhao, "Mathematical modeling of a pressurizer in a pressurized water reactor for control design," *Applied Mathematical Modelling,* vol. 65, pp. 187-206, 2019/01 2019, doi: 10.1016/j.apm.2018.08.006.

[9] P. Wang, J. Wan, S. Wu, and F. Zhao, "A fuzzy-logic-based pressure setpoint modification method for pressurized water reactor pressurizers," *Annals of Nuclear Energy,* vol. 135, p. 106954, 2020/01 2020, doi: 10.1016/j.anucene.2019.106954.

[10] C. Fazekas, G. Szederkényi, and K. M. Hangos, "A simple dynamic model of the primary circuit in VVER plants for controller design purposes," *Nuclear Engineering and Design,* vol. 237, no. 10, pp. 1071-1087, 2007/05 2007, doi: 10.1016/j.nucengdes.2006.12.002.

[11] S. K. Moghanaki and M. Rahgoshay, "Simulation of two-region and four-region models for typical PWR pressurizer and benchmark obtained results using available results," *Annals of Nuclear Energy,* vol. 63, pp. 302-308, 2014/01 2014, doi: 10.1016/j.anucene.2013.08.014.

[12] A. Pini, A. Cammi, L. Colombo, and A. B. Tigliole, "A non-equilibrium control oriented model for the pressurizer dynamics," *Progress in Nuclear Energy,* vol. 106, pp. 102-119, 2018/07 2018, doi: 10.1016/j.pnucene.2018.02.017.

[13] X. Zhong *et al.*, "Development of an improved non-equilibrium multi-region model for pressurized water reactor pressurizer," *Annals of Nuclear Energy,* vol. 126, pp. 133-141, 2019/04 2019, doi: 10.1016/j.anucene.2018.11.010.

[14] M. V. d. Oliveira and J. C. S. d. Almeida, "Application of artificial intelligence techniques in modeling and control of a nuclear power plant pressurizer system," *Progress in Nuclear Energy,* vol. 63, pp. 71-85, 2013/03 2013, doi: 10.1016/j.pnucene.2012.11.005.

[15] T. A. Mahmoud, A. A. Sheta, R. M. Fikry, E. H. Ali, S. M. El-araby, and M. I. Mahmoud, "Design of data-driven model for the pressurizer system in nuclear power plants using a TSK fuzzy neural network," *Nuclear Engineering and Design,* vol. 399, no. April, pp. 112015-112015, 2022, doi: 10.1016/j.nucengdes.2022.112015.

[16] L. N. F. Guimarães, N. d. S. Oliveira, and E. M. Borges, "Derivation of a nine variable model of a U-tube steam generator coupled with a three-element controller," *Applied Mathematical Modelling,* vol. 32, no. 6, pp. 1027-1043, 2008/06/01/ 2008, doi: https://doi.org/10.1016/j.apm.2007.02.022.

[17] P. Wang, X. Yan, and F. Zhao, "Multi-objective optimization of control parameters for a pressurized water reactor pressurizer using a genetic algorithm," *Annals of Nuclear Energy,* vol. 124, pp. 9-20, 2019/02 2019, doi: 10.1016/j.anucene.2018.09.026.

[18] H. Dubois, M.-A. Peraldi-Frati, and F. Lakhal, "A Model for Requirements Traceability in a Heterogeneous Model-Based Design Process: Application to Automotive Embedded Systems," presented at the 2010 15th IEEE International Conference on Engineering of Complex Computer Systems, 2010/03, 2010. [Online]. Available: http://dx.doi.org/10.1109/iceccs.2010.2.

[19] V. A. Vyawahare, G. Datkhile, P. Kadam, and G. Espinosa-Paredes, "Closed-loop controller design, stability analysis and hardware implementation for fractional neutron point kinetics model," *Nuclear Engineering and Technology,* vol. 53, no. 2, pp. 688-694, 2021/02 2021, doi: 10.1016/j.net.2020.07.026.

[20] Y. Ji, "The activation study of activated charcoal for radioactive off-gas treatment systems in NPPs," *Results in Engineering,* vol. 21, p. 101831, 2024/03/01/ 2024, doi: https://doi.org/10.1016/j.rineng.2024.101831.
[21] S. Alimah *et al.*, "Techno-economics of desalination cogeneration with SMR: Case study for prospective NPP in West Kalimantan," *Case Studies in Chemical and Environmental Engineering,* vol. 9, p. 100603, 2024/06/01/ 2024, doi: https://doi.org/10.1016/j.cscee.2023.100603.
[22] X. Chen, J. Coble, and F. Zhang, "A full-scope, high-fidelity simulator-based hardware-in-the-loop testbed for comprehensive nuclear power plant cybersecurity research," *Progress in Nuclear Energy,* vol. 175, p. 105348, 2024/10/01/ 2024, doi: https://doi.org/10.1016/j.pnucene.2024.105348.
[23] https://www.mathworks.com/?s_tid=gn_logo. (accessed.
[24] A. Ion., " GAOT-ECM: MATLAB Central File Exchange. Extension For Control And Modeling," *(https://www.mathworks.com/matlabcentral/fileexchange/51072-gaot-ecm-extension-for-control-and-modeling), ,* vol. Retrieved September 22, 2022., (2022).
[25] P. Wang *et al.*, "Control simulation and study of load rejection transient for AP1000," *Progress in Nuclear Energy,* vol. 85, pp. 28-43, 2015/11/01/ 2015, doi: https://doi.org/10.1016/j.pnucene.2015.05.008.
[26] S. Wu, P. Wang, H. Song, X. Wei, F. Zhao, and S. Revankar, "Modeling and load follow simulation of CPR1000 Nuclear Power Plant implementing Mechanical Shim control strategy," *Nuclear Engineering and Design,* vol. 352, p. 110161, 2019/10/01/ 2019, doi: https://doi.org/10.1016/j.nucengdes.2019.110161.
[27] Y. Qin, L. Sun, Q. Hua, and P. Liu, "A Fuzzy Adaptive PID Controller Design for Fuel Cell Power Plant," *Sustainability,* vol. 10, no. 7, p. 2438, 2018/07/12 2018, doi: 10.3390/su10072438.
[28] J. Espinosa, J. Vandewalle, and V. Wertz, "Fuzzy Logic, Identification and Predictive Control," in *Advances in Industrial Control*, ed: Springer London, 2005.
[29] V. A. Vyawahare and P. S. V. Nataraj, "Development and analysis of some versions of the fractional-order point reactor kinetics model for a nuclear reactor with slab geometry," *Communications in Nonlinear Science and Numerical Simulation,* vol. 18, no. 7, pp. 1840-1856, 2013/07/01/ 2013, doi: https://doi.org/10.1016/j.cnsns.2012.11.012.
[30] S. Motahhir, A. El Ghzizal, S. Sebti, and A. Derouich, "MIL and SIL and PIL tests for MPPT algorithm," *Cogent Engineering,* vol. 4, no. 1, p. 1378475, 2017/01/01 2017, doi: 10.1080/23311916.2017.1378475.
[31] K. N. Proskuryakov, M. S. Khvostova, R. M. Ismail, M. M. Kaverznev, and K. A. Yakovlev, "Application of information technology to create the digital acoustic models of VVER," *Nuclear Engineering and Design,* vol. 421, p. 113098, 2024/05/01/ 2024, doi: https://doi.org/10.1016/j.nucengdes.2024.113098.
[32] A. E. Salman and M. R. Roman, "A PWR core power control using optimized PID controller with SQP based on control rod positioning and variable coolant flow rate," *Annals of Nuclear Energy,* vol. 206, p. 110670, 2024/10/01/ 2024, doi: https://doi.org/10.1016/j.anucene.2024.110670.
[33] M. Hu, G. Zeng, H. Yao, and Y. Tang, "Processor-in-the-loop demonstration of coordination control algorithms for distributed spacecraft," presented at the The 2010 IEEE International Conference on Information and Automation, 2010/06, 2010. [Online]. Available: http://dx.doi.org/10.1109/icinfa.2010.5512151.

[34] P. Podder, M. Rubaiyat Hossain Mondal, and J. Kamruzzaman, "Chapter 1 - Iris feature extraction using three-level Haar wavelet transform and modified local binary pattern," in *Applications of Computational Intelligence in Multi-Disciplinary Research*, A. A. Elngar, R. Chowdhury, M. Elhoseny, and V. E. Balas Eds.: Academic Press, 2022, pp. 1-15.

[35] S. Saha, G. Saini, A. Chauhan, S. Upadhyay, R. Madurai Elavarasan, and M. S. Hossain Lipu, "Optimum design and techno-socio-economic analysis of a PV/biomass based hybrid energy system for a remote hilly area using discrete grey wolf optimization algorithm," *Sustainable Energy Technologies and Assessments,* vol. 57, p. 103213, 2023/06/01/ 2023, doi: https://doi.org/10.1016/j.seta.2023.103213.

[36] A. H. Khan and M. S. Islam, "A PCTRAN-BASED INVESTIGATION ON THE EFFECT OF INADVERTENT CONTROL ROD WITHDRAWAL ON THE THERMAL-HYDRAULIC PARAMETERS OF A VVER-1200 NUCLEAR POWER REACTOR," *Acta Mechanica Malaysia,* vol. 2, no. 2, pp. 32-38, 2019/10/10 2019, doi: 10.26480/amm.02.2019.32.38.

[37] T. A. Mahmoud, A. A. Sheta, R. M. Fikry, E. H. Ali, S. M. El-Araby, and M. I. Mahmoud, "Design of data-driven model for the pressurizer system in nuclear power plants using a TSK fuzzy neural network," *Nuclear Engineering and Design,* vol. 399, p. 112015, 2022/12/01/ 2022, doi: https://doi.org/10.1016/j.nucengdes.2022.112015.